SCHEME : K

Name : _____

Roll No. : _____ Year : 20___ 20___

Exam Seat No. : _____

LABORATORY MANUAL FOR

JAVA PROGRAMMING - 314317

COMPUTER ENGINEERING GROUP

MAHARASHTRA STATE BOARD OF
TECHNICAL EDUCATION, MUMBAI
(Autonomous) (ISO 9001: 2015) (ISO/IEC 27001:2013)

Vision

To ensure that the Diploma level Technical Education constantly matches the latest requirements of Technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

Mission

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the challenging technological & environmental challenges.

Quality Policy

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

Core Values

MSBTE believes in the following:

- Skill development in line with industry requirements
- Industry readiness and improved employability of Diploma holders
- Synergistic relationship with industry
- Collective and Cooperative development of all stake holders
- Technological interventions in societal development
- Access to uniform quality technical education

A Practical Manual
for

Java Programming

(314317)

Semester-IV

AI/ AN/ BD/ CM/ CO/ CW/ DS/ HA/ IF/ IH/ TE

Maharashtra State Board of Technical Education, Mumbai
(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)

'K' Scheme Curriculum

Maharashtra State Board of Technical Education, Mumbai
(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)
4th Floor, Government Polytechnic Building
49, Kherwadi, Bandra (East), Mumbai – 400051

Maharashtra State Board of Technical Education
Certificate

This is to certify that Mr./Ms. …… Roll No…… of the Fourth Semester of Diploma in …… Engineering/Technology (Program Code - .....4K) of the Institute …… (Inst. Code……) has completed the practical work satisfactorily for the course Java Programming(Course Code: 314317) for the academic year 20….. – 20…..... as prescribed in the curriculum.

Place ……. Enrollment No……

Date: …..... Exam Seat No. ……....

Course Teacher **Head of the Department** **Principal**

Seal of the Institute

Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much-needed industry relevant competencies and skills. Therefore, for the successful implementation of this curriculum, every practical has been designed to serve as a 'vehicle' to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'K' scheme laboratory manual development team designed the practicals to focus on outcomes, rather than the traditional age-old practice of conducting practical's to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read the concerned practical procedure that they will do the next day and understand minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. Students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

Java is one of the most popular programming languages used to create Web applications and platforms. It is designed to allow developers to write code that would run on any machine, regardless of architecture or platform. Diploma pass out should be able to use Object Oriented Programming construct of java and will be able to create the applications by using object- oriented concepts. They should possess basic skills of programming syntax and naming conventions. This course is designed to develop these vital skills in them through lab-based activities.

Although all care has been taken to check for mistakes in this laboratory manual, yet it is impossible to claim perfection especially as this is the first edition. Any such errors and suggestions for improvement can be brought to our notice and are highly welcome.

Program Outcomes (POs) to be achieved through Course:

PO1	Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
PO2	Problem analysis: Identify and analyses well-defined engineering problems using codified standard methods.
PO3	Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
PO4	Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
PO5	Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
PO6	Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
PO7	Life-long learning: Ability to analyses individual needs and engage in updating in the context of technological changes.

List of Relevant Skills

The following industry relevant skills of the competency "Apply advanced Java Programming Concepts" are expected to be developed in you by performing practical's of
this laboratory manual.

1. Develop standalone and network-based applications using Java.
2. Write and execute programs for web based as well as window-based applications.
3. Write and execute programs for networking.
4. Write and execute programs to store the data in database to perform Database
operations.

Practical Course Outcome Matrix

Course Outcomes (COs)

CO1	Develop java program using classes and objects.
CO2	Develop java program for implementing code reusability concept.
CO3	Develop program to implement multithreading and exception handling.
CO4	Develop java program for implementing event handling using window-based application Components
CO5	Implements network programming in java.
CO6	Develop java program for managing database.

Sr. No.	Title of the Experiment	CO1	CO2	CO3	CO4	CO5	CO6
1	* Setup Java Programming development environment using: • Command prompt.(Classpath and path setup) • Any IDE (Eclipse, Netbeans, VScode, Jcreator etc.).	✓					
2	Write programs to evaluate different types of expressions.	✓					
3	Write programs to demonstrate use of: if statements (all forms of • if statement • Switch – Case statement • Different types of Loops(for,while and do..while).	✓					
4	*Write programs for implementation of different methods of: • String class. • StringBuffer class.	✓					
5	* Write programs to demonstrate: • Use of Array. • Use of Vectors.	✓					
6	Write programs using Wrapper Class: • to convert primitive into object. • to convert object into primitive.	✓					
7	Develop a program for implementation of different types of constructors.	✓					
8	Develop program to implement: • Single inheritance. • Multilevel inheritance.		✓				
9	* Develop program for implementation of interface.		✓				
10	*Write programs to demonstrate use of: • Built in packages		✓				

	• User defined packages.						
11	*Write programs using multithreading.			✓			
12	Write programs for implementation of try, catch and finally block.			✓			
13	*Write programs for implementation of throw, throws clause.			✓			
14	* Write program to design any type of form using AWT components.				✓		
15	Write program to create a menu bar with various menu items and sub menu items.				✓		
16	Write program to demonstrate the use of border layout. The layout shows four buttons at four sides with captions “left”, “right”, “top” and “bottom” using Swing Components.				✓		
17	*Write program to design a calculator to demonstrate the use of grid layout using swing components				✓		
18	Write program using swing to display a JComboBox in a JFrame .				✓		
19	Write program to create JTree and JTable.				✓		
20	* Write program to handle key events and mouse events.				✓		
21	*Write program to implement action event in frame using swing components.				✓		
22	Write program to handle text event on swing components.				✓		
23	Write program to retrieve hostname and IP address using InetAddress class.				✓		
24	*Write program to demonstrate various methods of: • URL class. • URLConnection.				✓		
25	*Write program that demonstrates connection oriented communication using socket.					✓	
26	Write program to demonstrate sending and receiving data through datagram.				✓		
27	*Write program to: • Create sample database. • Make connectivity with database.					✓	
28	*Write program to implement following operations on database: • Insert record. • Update record. • Delete record.					✓	

29	Write program to demonstrate the use of PreparedStatement.					✓	
30	*Write program to retrieve data from table using ResultSet interface.(Use various methods of navigation methods).					✓	

Guidelines to Teachers

1. Teachers should align the explanation of the topic to teaching learning outcome (TLOs).
2. Refer to laboratory learning outcome (LLOs) for the execution of the practical to focus on the defined objectives.
3. Promote life-long learning by training the students to equip themselves with essential knowledge, skills and attitudes.
4. If required, provide demonstration for the practical emphasizing on the skills that the student should achieve.
5. Teachers should give opportunity to the students for exhibiting their skills after the demonstration.
6. Provide feedback and/or suggestions and share insights to improve effectiveness.
7. Assess students' skill achievement related to COs of each unit.

Instructions for Students

1. 100% attendance is compulsory for all practical sessions.
2. Students must adhere to ethical practices.
3. All the students must follow the schedule of practical sessions, complete the assigned work/activity and submit the assignment in stipulated time as instructed by the course teacher.
4. Students shall listen carefully the lecture given by teacher about importance of subject, learning structure, course outcomes.
5. Students shall understand the purpose of experiment and its practical implementation.
6. Students shall write the answers of the questions during practical.
7. Student should feel free to discuss any difficulty faced during the conduct of practical.
8. Students shall develop web based and window-based applications a expected by the industries.
9. Student shall attempt to develop related hands-on skills and gain confidence.
10. Students should develop habit to submit the write-ups on the scheduled dates and time.

Content Page

List of Practical and Formative Assessment Sheet

Sr. No	Practical Title	Date of Performance	Date of Submission	Assessment Marks (25)	Teacher's Sign	Remark
1	* Setup Java Programming development environment using: • Command prompt.(Classpath and path setup) • Any IDE (Eclipse, Netbeans, VScode, Jcreator etc.).					
2	Write programs to evaluate different types of expressions.					
3	Write programs to demonstrate use of: if statements (all forms of • if statement • Switch – Case statement • Different types of Loops(for,while and do..while).					
4	*Write programs for implementation of different methods of: • String class. • StringBuffer class.					
5	* Write programs to demonstrate: • Use of Array. • Use of Vectors.					
6	Write programs using Wrapper Class: • to convert primitive into object. • to convert object into primitive.					
7	Develop a program for implementation of different types of constructors.					
8	Develop program to implement: • Single inheritance. • Multilevel inheritance.					
9	*Develop program for implementation of interface.					

10	*Write programs to demonstrate use of: • Built in packages • User defined packages.					
11	*Write programs using multithreading.					
12	Write programs for implementation of try, catch and finally block.					
13	*Write programs for implementation of throw, throws clause.					
14	* Write program to design any type of form using AWT components.					
15	Write program to create a menu bar with various menu items and sub menu items.					
16	Write program to demonstrate the use of border layout. The layout shows four buttons at four sides with captions “left”, “right”, “top” and “bottom” using Swing Components.					
17	*Write program to design a calculator to demonstrate the use of grid layout using swing components					
18	Write program using swing to display a JComboBox in a JFrame					
19	Write program to create JTree and JTable.					
20	* Write program to handle key events and mouse events.					
21	*Write program to implement action event in frame using swing components.					
22	Write program to handle text event on swing components.					
23	Write program to retrieve hostname and IP address using InetAddress class.					

24	*Write program to demonstrate various methods of: • URL class. • URLConnection.					
25	*Write program that demonstrates connection oriented communication using socket.					
26	Write program to demonstrate sending and receiving data through datagram.					
27	*Write program to: • Create sample database. • Make connectivity with database.					
28	*Write program to implement following operations on database: • Insert record. • Update record. • Delete record.					
29	Write program to demonstrate the use of PreparedStatement.					
30	*Write program to retrieve data from table using ResultSet interface. (Use various methods of navigation methods).					
				Total		

***Total marks to be transferred to proforma published by MSBTE**

Note:

- '*' Marked Practical's (LLOs) Are mandatory.
- Minimum 80% of above list of lab experiment are to be performed.
- Judicial mix of LLOs are to be performed to achieve desired outcomes.

Practical No. 1: * Setup Java Programming development environment using: Command prompt. (Classpath and path setup),Any IDE (Eclipse, Netbeans, VScode, Jcreator etc.).

I Practical Significance

Java language is compiled and interpreted. Java is the popular platform, which is used to develop various applications for the systems as well as embedded devices like mobile, laptops, tablets & many more. It is an object-oriented programming language. Students will be able to setup Java environment for executing Java Programs, using command prompt using different IDEs like Eclipse, Jcreator, NetBeans and test the setup using small java programs.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 1.1 Install any IDE software application.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Java language is compiled and interpreted. Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

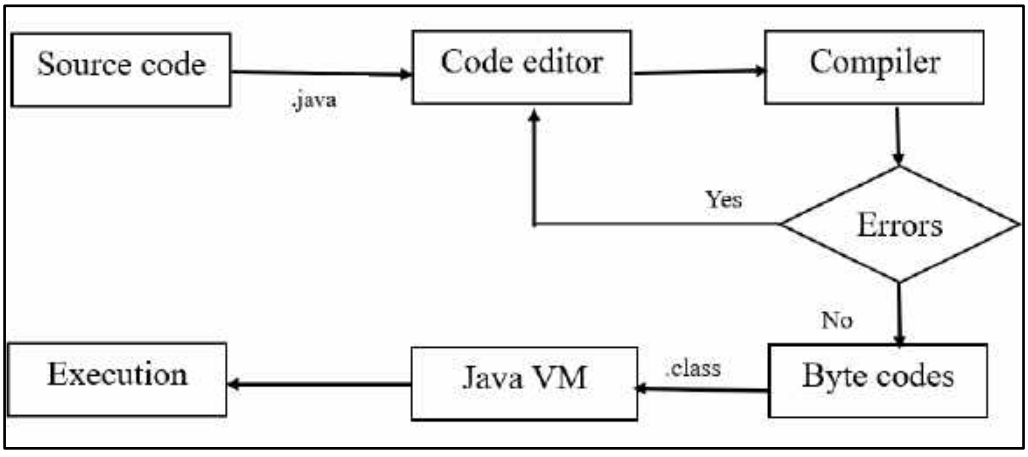


Fig no 1.1

- **Procedure:**

Installation for java Software:

1. Download JDK(jdk 1.8 Onwards) visit the

https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html

Download the windows version to suitable folder.

2. Double click the setup file.

3. Follow onscreen instructions.

4. When the setup is done, the complete screen appears, click on the 'Finish 'button. This completes the installation of JDK. To ensure the JDK installation / to determine the java version type the following command at the MS Dos prompt: <system prompt> java -version

It should show the output similar to following.

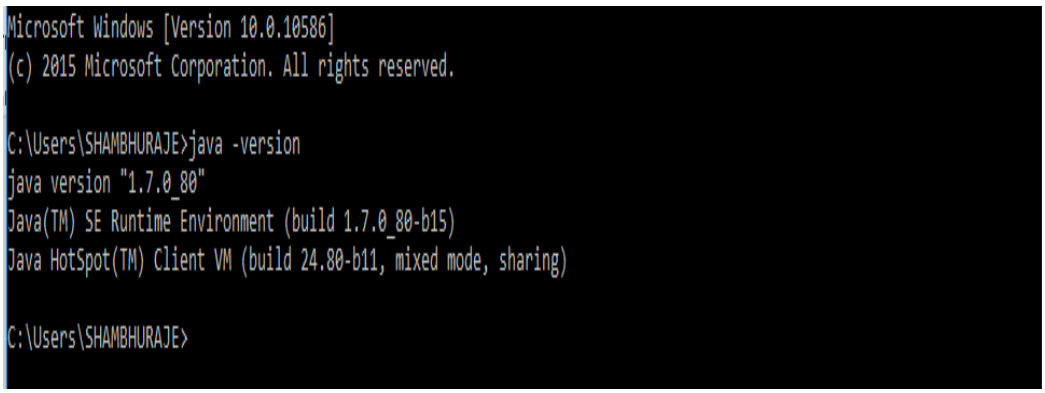


Fig no.1.2: Java Version

If not then set 'path' environment variable,

1. Go to File Explorer.
2. Right click on This PC.
3. Select advance system setting.
4. Click on environment variable.
5. Click on the edit.
6. Edit the System variable dialog box appears. In the variable value filed, append the path to the JDK bin directory (generally "C:\Program Files\Java\jdk1.7.0_80\bin". Click OK
7. Similarly set 'class path' environment variable

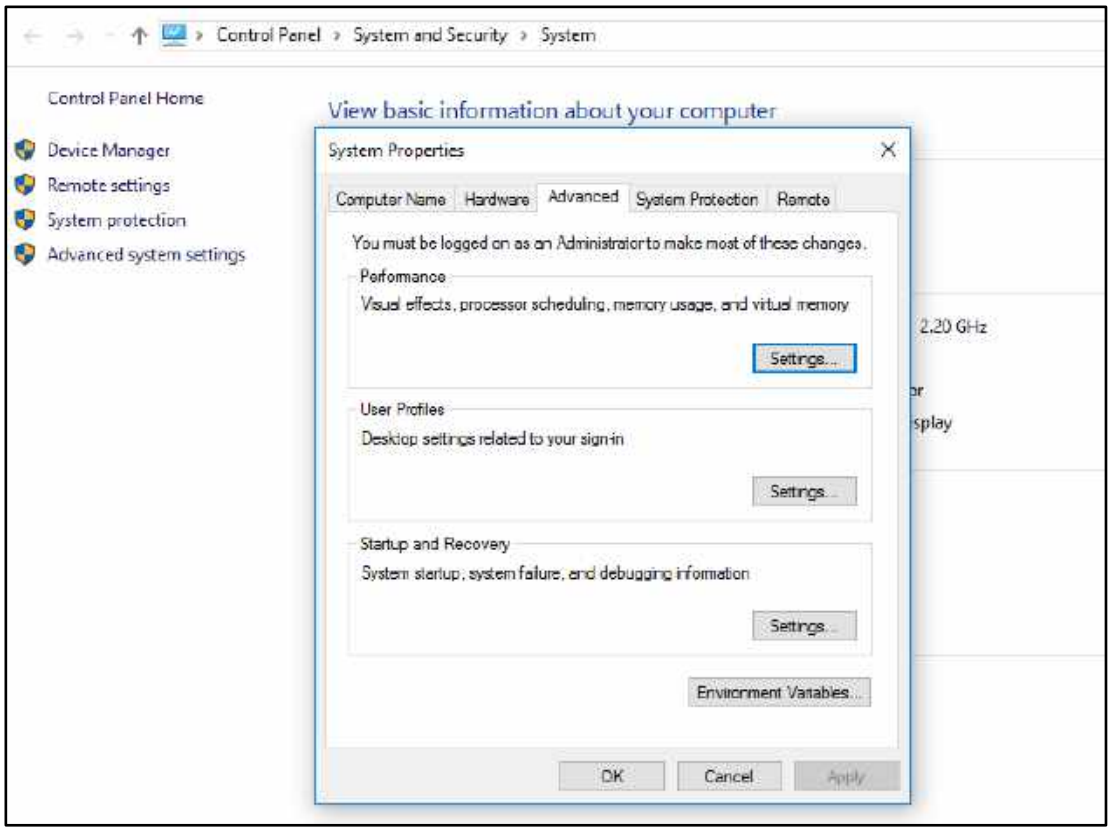


Fig no.1.3: Environment Available

Note: Follow the similar instructions for other platforms (say Unix, Linux, Mac) with appropriate jdk download.

- **Using an Eclipse:**

Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular.

- **Installing Eclipse IDE for Java Developers.**

To use Eclipse for Java Programming, you need to first install Java Development Kit(JDK)

1.Download Eclipse from
https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers

2. Install it(Follow onscreen instructions).

3. Testing setup using small programs:

Steps for editing and executing java programs:

- Using an editor (e.g. Notepad/ Notepad++)
    1. Open notepad
    2. Write the Java Program (called Java Source code) in notepad
    3. Save the file as **"Filename.java"** in some directory. The filename must be same as class name containing main() method.

4. Open MS-DOS Prompt.
5. Change directory containing to the program
6. Compile the program by using the command **javac <filename.java>**
7. Execute / Run the program by using the command java **<classname>**

- Using Eclipse
  1. Launch Eclipse by running "Eclipse.exe" from the Eclipse installed directory.
  2. Choose an appropriate directory for your workplace.
  3. To create a new Java Project using "File" menu =>" New" =>" Java Project"
  4. In "JRE", Select "Use default JRE". But make sure that your jdk is 1.8 and above.
  5. In "Project Layout" menu, select "Use project folder as root for sources and class files".
  6. Push "Finish: button.
  7. In the "Package Explorer (left pane)=>RIGHT -CLICK on "FirstProject=>New>Class
  8. Write a program
  9. Compile and execute program
  10. Observe output on the console panel.

- **Sample Program**

```
import java.io.*;
class Demo
{
  public static void main(String args[])
  {
  System.out.println("Welcome in JAVA Programming");
   }
}
```

- **Output**

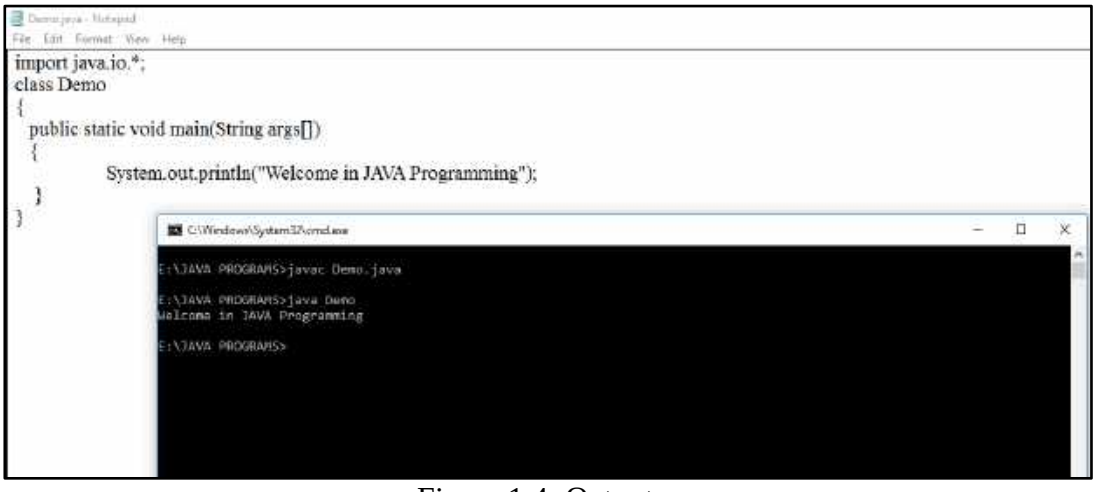


Fig no.1.4: Output

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write installation directory path of your directory?
2. Write value of path environment variable?
3. List folders created after installation.
4. Main Method is declared as static. Justify
5. Program is named with class containing main method. justfy.
6. Write the options provided by following JDK tools along with their use
   1.Java 2.javac 3.javadoc
7. List different versions of JDK
8. Test the setup using similar programs

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutorialspoint.com/java

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 2: Write programs to evaluate different types of expressions

I Practical Significance

An expression is a construct made up of variables, operators, and method invocations, which are constructed according to the syntax of the language, that evaluates to a single value.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 2.1 Implement programs to evaluate different types of Expressions.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Operator in java is a symbol that is used to perform opeartions. Eg. +,-,*,/,etc There are many types of operators in java which are given below

Operator Type	**Category**	**Precedence**
Unary	Postfix	*expr*++ *expr*--
	Prefix	++*expr* --*expr* +*expr* -*expr* ~ !
Arithmetic	Multiplicative	* / %
	Additive	+ -
Shift	Shift	<< >> >>>
Relational	Comparison	< > <= >= instanceof
	Equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	Ternary	? :
Assignment	Assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=
Special Operators	instanceof	Instanceof
	Member selection operator	.

Arithmetic Expressions:

An arithmetic expression is a combination of variables, constants, and operators arranged as per the syntax of the language. Java can handle any complex mathematical expressions. Remember java does not have an operator for exponentiation.

Algebraic expression	*Java expression*
a b−c	a*b−c
(m+n) (x+y)	(m+n)*(x+y)
$\frac{ab}{c}$	a*b/c
$3x^2+2x+1$	3*x*x+2*x+1
$\frac{x}{y}+c$	x/y+c

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What are operators in Java?
2. What is the arithmetic operator precedence in Java?
3. What are the comparative operators in Java?
4. What are expressions in programming?
5. What are unary and binary operators?

……

……

……

……

……

……

……
……
……
……
……
……
……

X References:

1. 1https://www.google.com/search?q=java+expressions+by+javatpoint
2. https://www.geeksforgeeks.org/what-is-an-expression-and-what-are-the-types-of-expressions/
3. https://codegym.cc/groups/posts/java-expressions-an-introduction-with-examples

   https://www.javatpoint.com/operators-in-java

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 3: Write programs to demonstrate use of: if statements (all forms of if statement Switch – Case statement Different types of Loops(for,while and do..while

I Practical Significance

In computer Science, Conditional statements, expressions and constructs are perform different computations or actions depending on whether boolean condition evaluates to true or false. Java Uses control statements to control the flow of execution of program based on certain conditions. A for loop is used to execute a block of code several times. Loop is used in programming to repeat a specific block of code until certain condition is true.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO.3.1 Develop program to implement different control structures.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Java Uses control statements to control the flow of execution of program based on certain conditions.

1. Java Selection Statements:

a. if
b. if-else
c. nested-if
d. if-else-if ladder

a. **if:** if statement is used to decide whether a certain statement or block of statements will be executed or not. i.e. if a certain condition is true then the block will be executed otherwise not

Syntax:

```
if(condition)
{
    // Statement to execute if the condition is true;
}
```

b. **if-else:** The if statement alone tells us that if a condition is true it will execute a block od statements and if the condition false, else block will be executed.

Syntax:

```
if(condition)
{       // Statement to execute if the condition is true;
}
else
{
        // Statement to execute if the condition is false;
}
```

c. **nested if:** A nested if is an statement that is the target of another if or else. Nested if statement means an if statement inside an if statement.

```
if(condition1)
{
        // Statement to execute if the condition is true;
        if(condition2)
{
        // Statement to execute if the condition is true;
}
}
```

d. **if-else-if ladder:** A user can decide among multiple options. The if statements are executed from top down. When one of condition is true, the statement associated with that if is executed , and the rest of the ladder is bypassed. If none of the condition is true ,then final else statement will be executed.

2. **Decision making using switch-case**

Syntax:

```
switch(expression)
{
case value1:
        //Statement
        Break;  // break is optional
        case value2:
         //Statement
        Break;  // break is optional
```

```
                                        .
                                        .
                                        .
        case valueN:
                    //Statement
                    Break;  // break is optional
            default:
            // Statements
}
```

3. **Conditional if(ternary operator):**

```
Syntax:
Result= testCondition ? Value1: Value 2;
```

4. **for Loop**

```
Syntax:
for(initialization; condition; increment/decrement)
{
    statements(s);
}
```

5. **while loop:**

```
Syntax:
while(condition)
{
    statements(s);
}
```

6. **do-while loop:**

```
Syntax:
        do
      {
            Statements;
        }while(condition);
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to check multiple conditions using if statement along with logical operators.
2. Write a program to check no is even or odd.
3. Write a program to check switch-case using character datatype.
4. Write a program to display 1 to 20 numbers using for, while and do-while loop.
5. Develop a program to use logical operators in do-while loop.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com/java-if-else
2. https://www.javatpoint.com/java-switch
3. https://www.javatpoint.com/java-for-loop
4. https://www.javatpoint.com/java-while-loop
5. https://www.javatpoint.com/java-do-while-loop

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 4: *Write programs for implementation of different methods of- String class, StringBuffer class.

I Practical Significance

String is a sequence of characters. Java String is a powerful concept. Students will be able to perform various operations on String object using different methods of String class.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 4.1 Develop program to implement different control structures.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Sr. No	Syntax	Task Performed
1	char charAt(int index)	It returns char value for the particular index
2	int length()	It returns string length
3	static String format(String format, Object... args)	It returns a formatted string.
4	static String format(Locale l, String format, Object... args)	It returns formatted string with given locale.
5	String substring(int beginIndex)	It returns substring for given begin index.
6	String substring(int beginIndex, int endIndex)	It returns substring for given begin index and end index.
7	boolean contains(CharSequence s)	It returns true or false after matching the sequence of char value.
8	static String join(CharSequence delimiter, CharSequence... elements)	It returns a joined string.
9	static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)	It returns a joined string.

10	boolean equals(Object another)	It checks the equality of string with the given object.
11	boolean isEmpty()	It checks if string is empty.
12	String concat(String str)	It concatenates the specified string.
13	String replace(char old, char new)	It replaces all occurrences of the specified char value.
14	String replace(CharSequence old, CharSequence new)	It replaces all occurrences of the specified CharSequence.
15	static String equalsIgnoreCase(String another)	It compares another string. It doesn't check case.
16	String[] split(String regex)	It returns a split string matching regex.
17	String[] split(String regex, int limit)	It returns a split string matching regex and limit.
18	String intern()	It returns an interned string.
19	int indexOf(int ch)	It returns the specified char value index.
20	int indexOf(int ch, int fromIndex)	It returns the specified char value index starting with given index.
21	int indexOf(String substring)	It returns the specified substring index.
22	int indexOf(String substring, int fromIndex)	It returns the specified substring index starting with given index.
23	String toLowerCase()	It returns a string in lowercase.
24	String toLowerCase(Locale l)	It returns a string in lowercase using specified locale.
25	String toUpperCase()	It returns a string in uppercase.
26	String toUpperCase(Locale l)	It returns a string in uppercase using specified locale.
27	String trim()	It removes beginning and ending spaces of this string.
28	static String valueOf(int value)	It converts given type into string. It is an overloaded method.

StringBuffer class Methods:

Sr. No	Syntax	Task Performed
1	public synchronized StringBuffer append(String s)	It is used to append the specified string with this string.
2	public synchronized StringBuffer insert(int offset, String s)	It is used to insert the specified string with this string at the specified position.
3	public synchronized StringBuffer replace(int startIndex, int endIndex, String str)	It is used to replace the string from specified startIndex and endIndex.
4	public synchronized StringBuffer delete(int startIndex, int endIndex)	It is used to delete the string from specified startIndex and endIndex.
5	public synchronized StringBuffer reverse()	is used to reverse the string.
6	public int capacity()	It is used to return the current capacity.
7	public char charAt(int index)	It is used to return the character at the specified position.
8	public int length()	It is used to return the length of the string i.e. total number of characters.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to show the use of all methods of String class.
2. Write a program to implement all methods of StringBuffer class.
3. List different constructors of StringBuffer class along with syntax
4. What is difference between = = ,equals() & compareTo()?

X References:

1. https://www.javatpoint.com/StringBuffer-class
2. https://www.javatpoint.com/java-string
3. https://www.w3schools.com/java/java_ref_string.asp

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
1.	Expected output:15%	
2.	Timely Submission:15%	
3.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 5: *Write programs to demonstrate: Use of Array. Use of Vectors.

I Practical Significance

Arrays store homogeneous data i.e. same type of data in consecutive memory locations which will help to fetch data in constant access time . Vector implements a dynamic array. Vector hold different number of objects. Students will be able to use Array & Vector in the program efficiently.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 5.1 Implement array and vectors in Java.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

- **Array:**

1. Creating an array:

```
dataType[] arrayRefVar;
or
dataType arrayRefVar[];
arrayRefVar = new type [size];
```

2.Array of Objects:

```
Class_name array_name=new class_name[size];
```

3.Types of arrays:

a) One Dimensional

```
eg. int [] intArrat =new int[20];
```

b) Multi Dimesional

```
eg. int[][] intArray= new int[10][20];  // a 2D array or matrix;
    int[][] intArray=new int[10][20][10]; // a 3D array;
```

- **Vector class methods: Vector** is like the *dynamic array* which can grow or shrink its size. Unlike array, we can store n-number of elements in it as there is no size limit. It is a part of Java Collection framework since Java 1.2. It is found in the java.util package and implements the *List* interface, so we can use all the methods of List interface here

Sr. No	Method	Description
1)	add()	It is used to append the specified element in the given vector.
2)	addAll()	It is used to append all of the elements in the specified collection to the end of this Vector.
3)	addElement()	It is used to append the specified component to the end of this vector. It increases the vector size by one.
4)	capacity()	It is used to get the current capacity of this vector.
5)	clear()	It is used to delete all of the elements from this vector.
6)	clone()	It returns a clone of this vector.
7)	contains()	It returns true if the vector contains the specified element.
8)	containsAll()	It returns true if the vector contains all of the elements in the specified collection.
9)	copyInto()	It is used to copy the components of the vector into the specified array.
10)	elementAt()	It is used to get the component at the specified index.
11)	elements()	It returns an enumeration of the components of a vector.
12)	ensureCapacity()	It is used to increase the capacity of the vector which is in use, if necessary. It ensures that the vector can hold at least the number of components specified by the minimum capacity argument.
13)	equals()	It is used to compare the specified object with the vector for equality.
14)	firstElement()	It is used to get the first component of the vector.
15)	forEach()	It is used to perform the given action for each element of the Iterable until all elements have been processed or the action throws an exception.
16)	get()	It is used to get an element at the specified position in the vector.
17)	hashCode()	It is used to get the hash code value of a vector.
18)	indexOf()	It is used to get the index of the first occurrence of the specified element in the vector. It returns -1 if the vector does not contain the element.

19)	insertElementAt()	It is used to insert the specified object as a component in the given vector at the specified index.
20)	isEmpty()	It is used to check if this vector has no components.
21)	iterator()	It is used to get an iterator over the elements in the list in proper sequence.
22)	lastElement()	It is used to get the last component of the vector.
23)	lastIndexOf()	It is used to get the index of the last occurrence of the specified element in the vector. It returns -1 if the vector does not contain the element.
24)	listIterator()	It is used to get a list iterator over the elements in the list in proper sequence.
25)	remove()	It is used to remove the specified element from the vector. If the vector does not contain the element, it is unchanged.
26)	removeAll()	It is used to delete all the elements from the vector that are present in the specified collection.
27)	removeAllElements()	It is used to remove all elements from the vector and set the size of the vector to zero.
28)	removeElement()	It is used to remove the first (lowest-indexed) occurrence of the argument from the vector.
29)	removeElementAt()	It is used to delete the component at the specified index.
30)	removeIf()	It is used to remove all of the elements of the collection that satisfy the given predicate.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to implement multidimensional array.
2. Write a program to display array elements using for-each loop.
3. Write a program to insert different elements in the Vector & display them
4. What is the use of new operator in defining an array?
5. Write a program to use different methods of Vector class.

X References:

1. https://www.javatpoint.com/array-in-java
2. https://www.javatpoint.com/java-string
3. https://www.w3schools.com/java/java_ref_string.asp

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 6: Write programs using Wrapper Class: to convert primitive into object. to convert object into primitive.

I Practical Significance

Wrapper classes are used to convert primitive data type into an object. The primitive data types are not objects. They are predefined in the language itself. Student should be able to use different wrapper classes and their methods.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 6.1 Convert primitive data types into object and vice-versa

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

- **Integer class methods:**

Sr. no	Modifier and Type	Method	Purpose
1	byte	byteValue()	It converts the given number into a primitive byte type and returns the value of integer object as byte.
2	Static int	compare()	It compares two int values numerically and returns the result in integer equivalent.
3	Int	compareTo()	It compares two integer objects numerically and returns the result as -1, 0 or 1.
4	double	doubleValue()	It converts the given Integer value and returns the result as a double equivalent.
5	boolean	equals()	It compares the value of the parameter to the value of the current Integer object and returns boolean (True or False).
6	float	floatValue()	It converts the given Integer value and returns the result as a float equivalent.
7	int	intValue()	It returns the value of the specified number as an int.
8	long	longValue()	It returns the value of the specified long object as long equivalent.

9	static int	parseInt()	It parses the String argument as a signed decimal Integer object.
10	short	shortValue( )	It returns the value of this Integer as a short type after a primitive conversion.
11	String	toString()	It returns a String object representing the value of the Number Object.
12	static Integer	valueOf()	It returns the relevant Integer Object holding the value of the argument passed.

Similar Wrapper class methods are available for Float, Short, Long, and Double class

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to show the use of Integer Wrapper class methods.
2. Write a different way to create object of the any primitive datatype.
3. Write methods of Number class to convert object into primitive datatypes.
4. Write a program to convert String value into Integer class object.
5. Write a program to make use of Character Wrapper class Methods.
6. Write a program to convert Integer object value into primitive datatype byte, short and double value

……

……

……

……

……

……

X References:

1. https://www.tutorialspoint.com/java/java_wrapper_classes.htm#:~:text=Wrapper%20classes%20are%20those%20whose,%2C%20Float%2C%20Long%2C%20Short.
2. https://www.javatpoint.com/wrapper-class-in-java
3. https://www.youtube.com/watch?v=Fyc86kVIePE
4. https://www.youtube.com/watch?v=9r1a8PzFXZA

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 7: Develop a program for implementation of different types of constructors.

I Practical Significance

A constructor in Java Programming is a block of code that initializes (constructs) the state and value during object creation. It is called every time an object with the help of a new () keyword is created. Even if you haven't specified any constructor in the code, the Java compiler calls a default constructor. There are two types of constructors in Java: no-arg(default) constructor, and parameterized constructor.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO1 - Develop java program using classes and objects.

IV Laboratory Learning Outcome(s)

LLO 7.1 Initialize objects using constructors

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

There are two types of constructors in Java

1. Default Constructor

Syntax:

```
Class_name()
{
        Statements for initialize the data members.
}
```

2. Parameterzied constructor

Syntax:

```
Class_name(parameter list)
{
        Statements for initialize the data members.
}
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Demonstrate use of at least two types of constructors.
2. How constructor overloading can be done?
3. Specify the situation when the default constructor is provided by the system.
4. What is the use of constructor in java?
5. Write a program to implement different types of constructors to perform addition of complex numbers?

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.geeksforgeeks.org/constructors-in-java/
2. https://www.javatpoint.com/java-constructor
3. https://www.w3schools.com/java/java_constructors.asp

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 8: Develop program to implement: Single inheritance. Multilevel inheritance.

I Practical Significance

Inheritance can be defined as the process where one class acquires the properties methods and fields of another. With the use of inheritance, the information is made manageable in a hierarchical order. The class which inherits the properties of other is known as sub class derived class, child class and the class whose properties are inherited is known as superclass base class, parent class. The student will be able to use different types of inheritance.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO2 - Develop java program for implementing code reusability concept.

IV Laboratory Learning Outcome(s)

LLO 8.1 Implement concepts of inheritance for code reusability.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

- **Inheritance:**

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

Types of Inheritance
1. Single Inheritance
2. Multiple Inheritance
3. Hierarchical Inheritance
4. Multiple Inheritance
5. Hybrid Inheritance

In java programming, multiple and hybrid inheritance is supported through interface only.

Defining a subclass:

```
class Subclass-name extends Superclass-name
{
        //methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to implement single inheritance.
2. Write a program to implement multilevel inheritance.
3. Develop a program to implement the multilevel inheritance.
4. Develop a program to calculate he room area and volume to illustrate the concept of single inheritance.

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

X References:

1. https://www.javatpoint.com/inheritance-in-java
2. https://www.geeksforgeeks.org/inheritance-in-java/
3. https://www.shiksha.com/online-courses/articles/multilevel-inheritance-in-java-blogId-157341

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 9: * Develop program for implementation of interface

I Practical Significance

Multiple inheritance in Java refers to a scenario where a class can inherit properties and methods from more than one superclass. However, Java does not support multiple inheritance directly through classes due to the complexity and ambiguity it can cause. The student will be able to use different types of inheritance.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO2 - Develop java program for implementing code reusability concept.

IV Laboratory Learning Outcome(s)

LLO 9.1 Implement multiple inheritance.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

1. Interfaces:

An interface in Java is a blueprint of a class. It has static constants and abstract methods. The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Defining Interfaces:

An interface is like a class. Interfaces also contain methods and variables but with a major difference. The difference is that interfaces define only abstract methods and final fields.

Interfaces do not specify any code to implement these methods & data fields contain only constants.

Syntax:

```
interface InterfaceName
{
        Variable declaration;
        Methods declaration;
}
```

Where interface is the keyword and InterfaceName is any valid java variable.
Variables are declared as:
static final type VariableName=Value;

Example: return_type methodName1(parameter_list)
Interface Definition:

```
interface Item
{
        static final int id=100;
        static final String name=”ABC”;
        void display();
}
```

Extending Interfaces:

Interfaces can be extended. The new sub interface will inherit all the members of the super interface.

```
interface name2 extends name1
{
                Body of name2;
}
```

Example:

```
interface ItemConstants
{
                int id=100;
                String name=”ABC”;  //All constants in one interface
        }
        interface Item extends ItemConstants
        {
                void display();  //All methods in other interface
        }
```

2. Multiple interface:

```
        interface ItemConstants
        {
                int id=100;
                String name=”ABC”;
        }
        interface ItemMethods
        {
                void display();
        }
        interface Item extends ItemConstants, ItemMethods
        {                       ……
```

……

}

3. Implementing Interfaces:

Interfaces are used like superclass whose features/properties are inherited by classes.It is mandatory to define a class that inherits the given interface.

```
class classname implements interfacename
{
        Body of classname
}
```

The class classname "implements" the interface interfacename.

```
class classname extends superclass implements interface1, interface2,……..
{
        Body of classname
}
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Demonstrate the use of interfaces to implement the concept of multiple inheritance.(Attach the code at the end).
2. Differentiate between class and interface.
3. Write similarities between interfaces and classes.
4. Write advantages of interfaces.
5. Develop a program to find area of rectangle and circle using interfaces.

……

……

……

……

……

X References:

1. https://www.geeksforgeeks.org/interfaces-in-java/
2. https://www.javatpoint.com/interface-in-java
3. https://www.youtube.com/watch?v=GLiqEppxtb4
4. https://www.youtube.com/watch?v=X9RNBbSK9Ms

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 10: *Write programs to demonstrate use of - Built in packages and User defined packages

I Practical Significance

Packages are collection of classes and interfaces. Importing of packages which helps in Reusability. Better organization of the classes and interfaces helps in resolving the name conflicts.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO2 - Develop java program for implementing code reusability concept.

IV Laboratory Learning Outcome(s)

LLO 10.1 Implement packages in Java

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

What is package?

It is a collection of similar types of classes, interfaces and sub-packages. Packages are categorized in two form, built-in package and user-defined package.

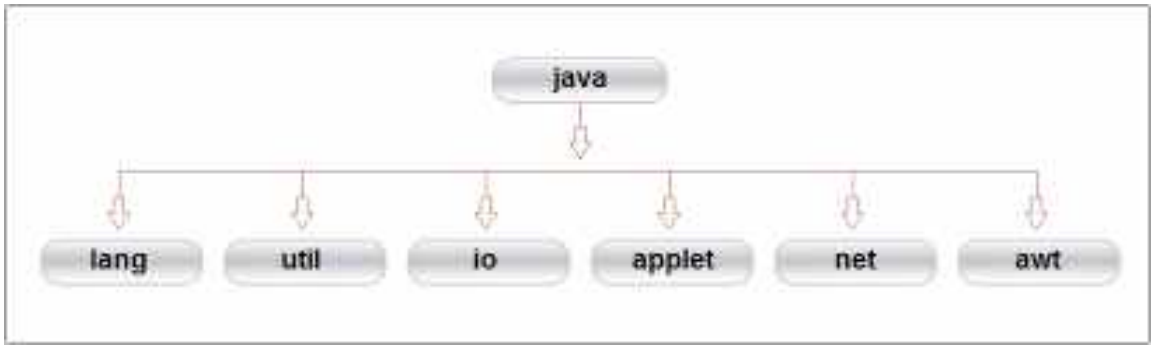


Fig.no.10.1: frequently used API Packages

Accessing the classes stored in a package:

Method 1: Using import Statements:

The first method is to use the fully qualified class name of the class:

import packagename.classname

Example:

import java.lang.Math

This statement imports the class Math and therefore class name can be used directly. It is not necessary to use the package name to qualify the class.

```
import java. lang.Math.sqrt(x);
```

Method 2: Shortcut Approach

```
import packagename.*;
import ava.lang.*;
```

This statement imports every class contained in the specified packages.
Above statement will bring all the classes of java.lang package

Creating packages

```
package PackageName;        //package declaration
public class FirstClass;    //class definition
{
        ……..                //(body of class)
}
```

Package Hierarchy

Package firstPackage.secondPackage;

Accessing a Package

A java system package can be accessed either using a fully qualified class name or using a shortcut approach through the import statement.

Syntax:

import package1[.package2][.package3].classname;

package1 is the name of the outer package, package2 is the name of the package that is inside the package1.

Package hierarchy consists of any number of packages. Finally the explicit classname is specified.

Example:

```
import firstPackage.secondPackage.MyClass;      // fully qualified class name
OR
import packagename.*;
import firstpackage.*;          //shortcut Approach
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a program to implement user defined packages in terms of creating a new package and importing the same.
2. Define a package named myInstitute include class named as department with one method to display the staff of that department. Develop a program to import this package in a java application and call the method defined in the package.
3. Develop a program which consists of the package named let_me_calculate with a class named Calculator and a method named add to add two integer numbers. Import let_me_calculate package in another program (class named Demo) to add two numbers.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.geeksforgeeks.org/packages-in-java/
2. https://www.javatpoint.com/packagese-in-java

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 11: *Write programs using multithreading

I Practical Significance

Multithreading technique in java helps to run multiple programs or a processes concurrently by utilizing the maximum CPU time. Multithreaded technique is implemented by creating, extending, implementing by thread. Student will be able to implement different types of thread methods by assigning the priority illustrate simultaneous execution of thread operation

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO3- Develop program to implement multithreading and exception handling.

IV Laboratory Learning Outcome(s)

LLO 11.1 Execute different processes simultaneously using multithreading

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Multithreading:

Multithreading is a small program (process) which is divided into two or more subprogram (processes), that can be implemented simultaneously.

Thread: It is a small process which is used to divide a program into number of sub parts and each part can be executed in parallel.

Creating threads:

Threads are implemented using a method called run().

Syntax:

```
public void run()
{
        ……..
        ……..
        ……
}
```

A Thread can be created in two ways:

1. By creating a thread class:
2. By converting a class to a thread

1. By creating a thread class:

Define a class that extends Thread class and override its run()

Extending the thread Class

```
class MyThread extends Thread
{
        ……
        ……
        ……
}
```

2. By converting a class to a thread:

Define a class that implements Runnable interface. The Runnable interface has only one method,run()

```
class A implements Runnable
{
        ……..
        ……..
}
```

Synchronization

It means only a single thread can execute a block of code at the same time.

Example :

The method that will read information from a file and the method that will update the same file may be declared as synchronized.

synchronized void update()

Java Thread Methods:

Sr.No	Type	Method	Description
1	void	start()	It is used to start the execution of the thread.
2	void	run()	It is used to do an action for a thread.
3	static void	sleep()	It sleeps a thread for the specified amount of time.
4	static Thread	currentThread()	It returns a reference to the currently executing thread object.
5	void	join()	It waits for a thread to die.
6	Int	getPriority()	It returns the priority of the thread.
7	void	setPriority()	It changes the priority of the thread.
8	String	getName()	It returns the name of the thread.
9	void	setName()	It changes the name of the thread.
10	Long	getId()	It returns the id of the thread.
11	Boolean	isAlive()	It tests if the thread is alive.
12	static void	yield()	It causes the currently executing thread object to pause and allow other threads to execute temporarily.
13	Void	suspend()	It is used to suspend the thread.

14	Void	resume()	It is used to resume the suspended thread.
15	Void	stop()	It is used to stop the thread.
16	Void	destroy()	It is used to destroy the thread group and all of its subgroups.
17	String	toString()	It is used to return a string representation of this thread, including the thread's name, priority, and thread group.
18	void	notify()	It is used to give the notification for only one thread which is waiting for a particular object.
19	void	notifyAll()	It is used to give the notification to all waiting threads of a particular object.
20	String	toString()	It is used to return a string representation of this thread, including the thread's name, priority, and thread group.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Develop a simple real-life application program to illustrate the use of multithreads.
2. Explain the use of keyword synchronized.
3. Differentiate between notify() and notifyAll()?
4. Implement multithreading to perform simultaneous processes.
5. Create three threads and run these threads according to setPriority.

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com/multithreading-in-java
2. https://www.tutuorialspoint.com/multithreading-in-java

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 12: Write programs for implementation of try, catch and finally block

I Practical Significance

Managing errors and Exception handling helps to detect exceptional conditions in a program and fix the exceptions as and when they occur. Student will be able to handle different types of exceptions using try, catch and finally blocks.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO3- Develop program to implement multithreading and exception handling.

IV Laboratory Learning Outcome(s)

LLO 12.1 Identify the different types of errors using exception handling

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Errors:

Errors are the mistakes that can make a program go wrong.

Error may produce a wrong results or abruptly terminates the execution of the program or may cause the system to crash.

Detecting and managing the errors is very important during the program execution.

Types or Errors:

1. Compile-time errors
2. Run-time errors

Exception Handling Tasks:

` 1. Find the Problem(Hit the exception)
2. Inform that an error has occurred(Throw the exception)
3. Receive the error information(Catch the exception)
4. Take corrective actions(Handle the exception)

Common Java Exception

Sr. No	Exception Type	Cause of Exception
1	ArithmeticException	Caused by math errors such as division by zero
2	ArrayIndexOutOfBoundsException	Caused by bad array indexes
3	ArrayStoreException	Caused when a program tries to store the wrong type of data in an array

4	FileNotFoundException	Caused by an attempt to access a non-existent file
5	IOException	Caused by general I/O failures, such as inability to read from a file
6	NullPointerException	Caused by referencing a null object
7	NumberFormatException	Caused when a conversion between strings and number fails
8	OutOfMemoryException	Caused when there is not enough memory to allocate a new object
9	SecurityException	Caused when an applet tries to perform an action not allowed by the browser's security setting
10	StackOverflowException	Caused when the system runs out of stack space
11	StringIndexOutOfBoundsException	Caused when a program attempts to access a non-existent character positions in a string.

Categories of Exceptions:

1. **Checked Exceptions**

   Checked exceptions are explicitly handled in the code itself using try catch blocks.

   These are extended from the java.lang.Exceptions class

2. **Unchecked Exceptions**

   Unchecked exceptions are not necessarily handled in the program code , instead the JVM handles such exceptions.

Exceptions Handling Code:

try:

A keyword 'try' is used for a block of code that causes an error condition and 'throw' an exception

catch:

A keyword 'catch' is used for a block of code that 'catches' the exception thrown by the 'try'

block and handles it properly.

Syntax: Simple try , catch and finally statement

```
try
{
        Statement;              //generates an exception
}
catch(Exception-type e)
{
        Statement;              //processes the exception
}
……..
finally
{
        ……. // optional
```

}

finally Statement:

finally statement handles an exception that is not caught by any of the earlier catch statements. It is used to handle any exception generated within a try block.It is written immediately after the try block or after the last catch block.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Demonstrate exception handling using try , catch and finally block.
2. Differentiate between error and exception in java.
3. Can we throw exception manually? Illustrate with sample program \
4. Develop a program to accept a password from the user and throw "Authentication Failure" Exception if the password is incorrect.
5. How exception is thrown by main method?

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 13: *Write programs for implementation of throw, throws clause

I Practical Significance

The throw and throws clause is used to explicitly throw an exception from a method or any block of code. It is mainly used for throwing the custom exceptions(User defined exceptions). Students will be able to throw the user defined exceptions

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO3- Develop program to implement multithreading and exception handling.

IV Laboratory Learning Outcome(s)

LLO 13.1 Manage different types of user defined exceptions

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Throwing Our own Exceptions:

throw: All methods in java use the 'throw' statement explicitly to **throw** an exception from a method or any block of code. The **throw** is keyword in java. **throw** can be used for either checked or unchecked exception. It is mainly used to **throw** custom exceptions.

Synatx:

throw Throwable-instance;

where **Throwable-instance** must be an object of type Throwable or a subclass of Throwable.

throws: It is a keyword in java.It is used to declare an exception . It is used with a method signature. Mutliple exceptions can be declared through throws.

Syntax:

type method_name (parameters) throws exception list

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Demonstrate use of throw and throws clause.
2. State the situation where throws clause is used.
3. Write the simple program for throwing our own exceptions.
4. Define an exception called "NotMatchExceptons" that is thrown when a string is not equals to "India". Write a program that uses this exceptions.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 14: * Write program to design any type of form using AWT

I Practical Significance

Text Field, Text Area, Button, Checkbox, Radio Buttons(Check Box Group) are the AWT components. Used to design the GUI in java A component is object having representation that can be displayed on the screen to interact with the user. The List and Choice components lets the user choose one option from list of available options. A choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one time may be selected from a Choice. A list may be selected from a choice. A list may be displayed in such a way that several List items are visible. A list supports the selection of one or more List items.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4- Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 14.1 Design GUI using different AWT components.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

AWT is a java programming language class library. Components are visible objects that can be interact with the user. Containers(Frame, Panel, Applet) are used to hold components using in a specific layout.

Using applet window, design following AWT components using add() method of components class. Following are some AWT components.

1. Label: Creates a label that displays a string.
2. TextField creates and accepts a single-line text from user.
3. TextArea creates and accepts multiline text from user.
4. Button creates a push button
5. Checkbox creates a checkbox which is used to select multiple options.
6. CheckboxGroup creates a group of checkbox to act as radio button.

To Create textArea

TextArea ta=new TextArea(String str, int nooflines)

To create RadioButton(CheckBoxGroup)

```
CheckBox cb1,cb2;
CheckBoxGroup cbg;
cb1= new CheckBox("Male", cbg, true)
cb2= new CheckBox("Female", cbg, false)
```

List: Creates a list from which the user can be choose list items.

Constructors:

List() // allows only one item to be selected

List(int numRows) // no of entries will always be visible.

List(int numRows, Boolean multiple Select)// if it is true then user select multiple items . If te is false then only item may be selected.

Choice

The Choice class is used to create a pop-up list of items from which the user may choose. When the user clicks on it. The whole list of choices pops up and new selection can be made. Choice defines the default constructor , which creates an empty list. To add a selection to the list call add()

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Design an applet/application to demonstrate the use of Radio Button and Checkbox.
2. Design an applet/application to create form using Text Filed, Text Area, Button and Label
3. Write a program to create three Buttons with Caption OK RESET and CANCEL
4. Develop an applet/application using List components to add names of 10 different cities.
5. Develop applet/application to select multiple names of news papers.

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 15: Write program to create a menu bar with various menu items

I Practical Significance

The Menu Bar lets the user to select different menu options from the list of available menu items and perform operations on the selected menu item.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 15.1 Design GUI using different menu class.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

A top –level window can have a menu bar associated with it. A menu bar displays a list of top-level menu choices. Each choice is associated with a drop-down menu.

This concept is implemented in the AWT by the following classes:

Menu bar, Menu and Menu Item

To create a menu bar ,first create an instance of Menu Bar. His class Only defines the default constructor. Next , create instances of Menu that will define the selections displayed on the bar.

Menu() //creates an empty menu

Menu(String optionName) // name of the menu selection

Menu(String optionName, Boolean removable) // If removable is true , the menu can be removed and allowed to float free. Otherwise , it will remain attached to the manu bar.

Individual menu items are of type MenuItem . It Defines these constructors:

1. MenuItem()
2. MenuItem(String itemName) // the name shown in the menu
3. MenuItem(String itemName, MenuShortcut keyAccel) // keyAccel is the menu shortcut for this item.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Design an applet/application to demonstrate the use of Radio Button and Checkbox.
2. Write a program which creates Menu of different colors and disable menu item for Black color.
3. Write the procedure to assign shortcut key to the Menu Item

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 16: Write program to demonstrate the use of border layout. The layout shows four buttons at four sides with captions “left”, “right”, “top” and “bottom” using Swing Components.

I Practical Significance

A layout manager automatically arranges your controls within a window. While it is possible to layout Java controls by hand, too, you generally won’t. It is very tedious to manually lay out a large number of components

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 16.1 Design GUI using border layout manager.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Layout Manager is a facility that determines how components should be arranged when they are added to the container. Layout Manager is an interface that is implemented by all the classes of layout managers. There are following classes that represent the layout managers.

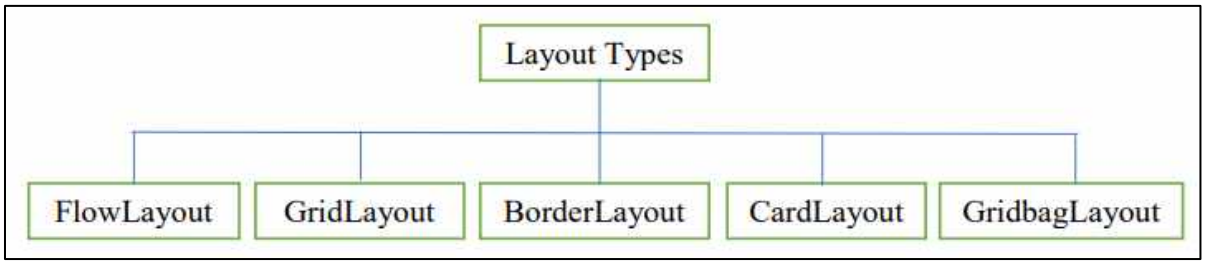


Fig no.:16.1

Understand the default layout for different containers such as Applet, Frame, Panel. The BorderLayout is used to arrange the components in five regions: north, south, east, west, and center. Each region (area) may contain one component only. It is the default layout of a frame or window.

The BorderLayout provides five constants for each region:

- public static final int NORTH
- public static final int SOUTH
- public static final int EAST

- public static final int WEST
- public static final int CENTER

Constructors of BorderLayout class:

- BorderLayout(): creates a border layout but with no gaps between the components.
- BorderLayout(int hgap, int vgap): creates a border layout with the given horizontal and vertical gaps between the components.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

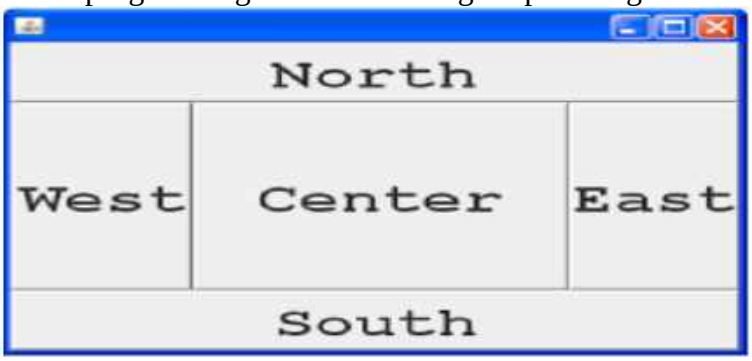
……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Give name of default Layout for Different container.
2. List the names of BorderLayout regions.
3. Write the use of Insets in border layout
4. Write a program to generate following output using Border Layout



……

……

……

……

……
……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 17: *Write program to design a calculator to demonstrate the use of grid layout using swing components.

I Practical Significance

A layout manager automatically arranges your controls within a window. While it is possible to lay out Java controls by hand, too, you generally won't. It is very tedious to manually lay out a large number of components

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 17.1 Design GUI using grid layout manager

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

The Java GridLayout class is used to arrange the components in a rectangular grid. One component is displayed in each rectangle.Grid Layout is used to make a bunch of components equal in size and displays them in the requested number of rows and columns. One component is displayed in each rectangle.

The list of Constructor for GridLayout are:

1. GridLayout(): creates a grid layout with one column per component in a row. The GridLayout() constructor creates only one row.
2. GridLayout(int rows, int columns): creates a grid layout with the given rows and columns but no gaps between the components.
3. GridLayout(int rows, int columns, int hgap, int vgap): creates a grid layout with the given rows and columns along with given horizontal and vertical gaps if we give setLayout(null) the default layout is disabled. Then we have to use setBounds method to layout the components.

Commonly Used Methods:

1. addLayoutComponent(String str, Component cmp): Adds the specified component with the specified name to the layout.
2. setColumns(int cl): Sets the number of columns in this layout to the specified value.
3. setHgap(int hgap): Sets the horizontal gap between components to the specified value.

4. setRows(int rw): Sets the number of rows in this layout to the specified value.
5. setVgap(int vgap): Sets the vertical gap between components to the specified value.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write java Program to Demonstrate Grid of 5* 5
2. Write a program to display The Number on Button from 0 to 9.
3. Write a program to generate following output

Fig no.:17.1

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 18: Write program using swing to display a JComboBox in a JFrame.

I Practical Significance

The Swing Components are very useful to design interactive application and it provide rich look and feel to the components. Swing components are light weight and platform independent. Swing supplies additional controls such as TabbedPane, ScrollPane, Trees and Tables. JFrame and JApplets are used to design windows and web applications

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 18.1 Implement swing components in a frame.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

JApplet, an extension of Applet, is the foundation of swing. The root pane, glass pane, and content pane are just a few of the panes that JApplet provides. Call the JApplet object's add () function for the content pane when adding a component to an instance of the JApplet.

A TextField and a dropdown list are combined to create a JComboBox. JComponent is an alias of JComboBox. Though it can also provide a drop-down list with options for the user to choose from, it typically shows a single entry. The popup menu of choices is displayed using the Choice class object. The option that the user has chosen appears at the top of a menu. It is descended from JComponent.

Constructors of JComboBox

1. JComboBox():Creates a JComboBox with a default data model.
2. JComboBox(Vector v): Creates a JComboBox that contains the elements in the specified Vector.
3. JComboBox(Object[] items): Creates a JComboBox that contains the elements in the specified array.

Commonly used Methods:

1. addItem(Object anObject):It is used to add an item to the item list.
2. removeItem(Object anObject):It is used to delete an item to the item list.
3. removeAllItems():It is used to remove all the items from the list.

4. getItemAt(int i) : returns the item at index i
5. getSelectedItem() : returns the item which is selected.
6. showPopup() : causes the combo box to display its popup window.
7. setSelectedItem(Object a): sets the selected item in the combo box display area to the object in the argument.
8. setPopupVisible(boolean v): sets the visibility of the popup.
9. setMaximumRowCount(int count): sets the maximum number of rows the JComboBox displays.
10. setEnabled(boolean b): enables the combo box so that items can be selected.
11. removeItem(Object anObject) : removes an item from the item list.
12. removeAllItems(): removes all items from the item list.
13. getItemCount() : returns the number of items in the list.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the difference between AWT and Swing
2. State the features of Swing
3. Name the method to obtain ContentPane in swing.
4. Write a program code to generate the following output

Fig no.:18.1

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 19: Write program to create JTree and JTable.

I Practical Significance

Tree is used to represent the hierarchical view of the data. The tree control shows the data in treelike structure such as root and its leaves. A table is a component that displays rows and columns of data. Students will be able to use JTable class in Java which extends JComponent. The cursor can be dragged on column boundaries to resize column.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 19.1 Design tree and table using advanced swing components in a frame.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

JTree

JTree is a complex component. It has a 'root node' at the top most which is a parent for all nodes in the tree. It inherits JComponent class.

Constrtuctors of JTree class

- JTree(Hashtable ht) : The first form creates a tree in which each element of the hash table ht is a child node.
- JTree(Object obj[]): Each element of the array obj is a child node.
- JTree(TreeNode tn):The tree node tn is the root of the tree.
- JTree(Vector v):It uses the elements of vector v as child nodes.

Various Methods of JTree class:

- getPathForLocation(): It is used to translate a mouse click on a specific point of the tree to a tree path.

Syntax : TreePath getPathForLocation(int x,int y)

where (x,y) are the coordinates at which the mouse is clicked. The return value is as TreePath object that encapsulates information about the tree node that was selected by the user.

- TreeNode interface :It declares methods that obtain information about a tree node.
- MutableTreeNode interface : It extends TreeNode.It declares methods that can insert and remove child nodes or change the parent node.

- DefaultMutableTreeNode class implements the MutableTreeNode interface.It represents a node in a tree.

DefaultMutableTreeNode(Object obj):Here,obj is the object to be enclosed in this tree node. The new tree node doesn't have a parent or children. To create hierarchy of tree nodes the add() can be used as void add(MutableTreeNode child) :added child to the current node.

JTable

Tables are implemented by the JTable class, which extends JComponent.

Constructors of JTable:

1. JTable() : Creates a table with empty cells
2. JTable(Object data[ ][ ], Object colHeads[ ]): Here, data is a two-dimensional array of the information to be presented, and colHeads is a one-dimensional array with the column headings.

Here are the steps for using a table in an applet:

1. Create a JTable object.
2. Create a JScrollPane object. (The arguments to the constructor specify the tableand the policies for vertical and horizontal scroll bars.)
3. Add the table to the scroll pane.
4. Add the scroll pane to the content pane of the applet

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the use of JTree in Swing.
2. State the use of getPathForLocation() method.
3. List different packages used to implement JTree programs.
4. Name the superclass of JTable component.
5. Write a Jtree program to show root directory and its subFolders of your System.

6. Write a Java program to create a table of Name of Student, Percentage and Grade of 10 students using JTable.

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 20: * Write program to handle key events and mouse events.

I Practical Significance

A KeyEvent is generated when keyboard input occurs. When key is pressed, released or typed, key event is generated. Students will be able to understand the method to register an object and handle various key of Keyboards. The MouseListener and MouseMotionListener interface are used to implement the different types of events that are created when the mouse is clicked, dragged, dropped, released, entered and exited in a component. These two interfaces let the user to handle the events and perform some action in that event.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 20.1 Implement various keys and mouse events.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

- KeyEvent

When keyboard input is occurred, a KeyEvent is generated. There are mainly three types of key events that are recognized by integer constants as follows:

KEY_PRESSED, KEY_RELEASED, KEY_TYPED

When key is pressed or released, first two events are generated. When character is pressed, the last event occurs. All Keys do not generate character. E.g. Shift Key InputEvent is super class of KeyEvent. The class which processes the KeyEvent should implement KeyListener interface. The object of that class must be registered with a component. The object can be registered using the addKeyListener () method.

Methods of KeyListener interface:

void keyPressed(KeyEvent e) : Invoked when a key is pressed

void keyReleased(KeyEvent e) : Invoked when a key has been released

void keyTyped(KeyEvent e) : Invoked when a key has been type

- MouseEvent

The java's MouseListener interface allows the user to handle the events when the user performs some events using mouse.The events are MouseClicked, MousePressed, MouseEntered, MouseExited, and Mouse released.

The list of available methods are:

1. void mouseClicked(MouseEvent me)
2. void mouseEntered(MouseEvent me)
3. void mouseExited(MouseEvent me)
4. void mousePressed(MouseEvent me)
5. void mouseReleased(MouseEvent me)

In the similar way we can handle the events when mouse is moved or dragged. The MouseMotion Interface defines the following methods.

1. void mouseDragged(MouseEvent me)
2. void mouseMoved(MouseEvent me)

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the situation when all three events of KeyListener interface are generated?
2. List various methods of MouseListener and MouseMotionListener.
3. Write a program to generate KeyEvent when a key is pressed and display "Key Pressed" message
4. Write a program to demonstrate the use of mouseDragged and mouseMoved method of MouseMotionListener

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 21: *Write program to implement action event in frame using swing components.

I Practical Significance

The ActionEvent is generated when button is clicked or the item of a list is double-clicked. This method is public as an implementation side-effect. Handles the action Performed event by invoking the actionPerformed methods on listener. Processes action events occurring on this menu item, by dispatching them to any registered

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 21.1 Implement action event in java.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event package. It has only one method: actionPerformed()

- actionPerformed() method

The actionPerformed() method is invoked automatically whenever you click on the registered component.

public abstract void actionPerformed(ActionEvent e);

The common approach is to implement the ActionListener. If you implement the ActionListener class, you need to follow 3 steps:

1) Implement the ActionListener interface in the class:

public class ActionListenerExample Implements ActionListener

2) Register the component with the Listener:

component.addActionListener(instanceOfListenerclass);

3) Override the actionPerformed() method:

```
public void actionPerformed(ActionEvent e)
        {
         }
```

- Using Anonymous class

We can also use the anonymous class to implement the ActionListener.

```
b.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent e){
    tf.setText("Welcome ");
}
});
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is actionPerformed in Java?
2. What is the difference between action listener and event listener?
3. What is the ActionListener class?
4. Write Java Program to display following output.

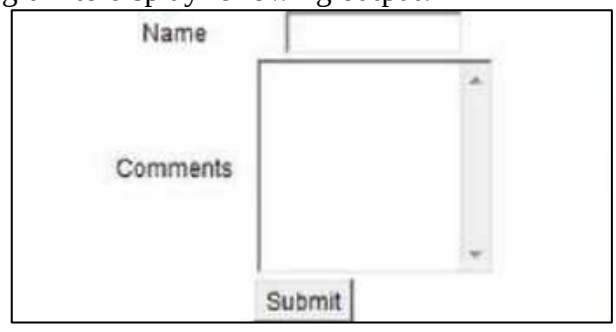


Fig no.: 21.1

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 22: Write program to handle text event on swing components.

I Practical Significance

It is commonly used to trigger actions in a GUI application. A text component is a self-contained editor with which you can write anything you want on your page. The Text Field control is used to accept the input from user. An interface is used to implement the methods as per the users' requirements and achieve the concept of multiple inheritance in java

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 22.1 Implement text event in java .

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

JTextField is a part of javax.swing package. The class JTextField is a component that allows editing of a single line of text. JTextField inherits the JTextComponent class and uses the interface SwingConstants.

- The constructor of the class are :
  - JTextField() : constructor that creates a new TextField
  - JTextField(int columns) : constructor that creates a new empty TextField with specified number of columns.
  - JTextField(String text) : constructor that creates a new empty text field initialized with the given string.
  - JTextField(String text, int columns) : constructor that creates a new empty textField with the given string and a specified number of columns .
  - JTextField(Document doc, String text, int columns) : constructor that creates a textfield that uses the given text storage model and the given number of columns.
- Methods of the JTextField are:
  - setColumns(int n) : set the number of columns of the text field.

- setFont(Font f) : set the font of text displayed in text field.
- addActionListener(ActionListener l) : set an ActionListener to the text field.
- int getColumns() : get the number of columns in the textfield.

```
import java.awt.event.*;
import javax.swing.*;
class text extends JFrame implements ActionListener {
        static JTextField t;
        static JFrame f;
        static JButton b;
        static JLabel l;
        text()
        {
        }
        public static void main(String[] args)
                        {
                                f = new JFrame("textfield");
                                l = new JLabel("nothing entered");
                                b = new JButton("submit");
                                text t = new text();
                                b.addActionListener(te);
                                t = new JTextField(16);
                                JPanel p = new JPanel();
                                p.add(t);
                                p.add(b);
                                p.add(l);
                                f.add(p);
                                f.setSize(300, 300);
                                f.show();
                        }
                        public void actionPerformed(ActionEvent e)
                        {
                                String s = e.getActionCommand();
                                if (s.equals("submit")) {
                                        l.setText(t.getText());
                                        t.setText(" ");
                                }
                        }
                }
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Which component can be used to accept the multiline input from user.
2. Which class is the Swing version of a text field?
3. What is the difference between Swing text field and text area?
4. Write a program using JTextField to perform the addition of two numbers.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 23: Write program to retrieve hostname and IP address using InetAddress class.

I Practical Significance

Java provides support for communication between two or more computers by the way of socket programming. It provides the classes for both the protocols that is UDP and TCP. Socket lets the user to create a client server communication in the network and share the data/information in it

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 23.1 Extract the hostname and IP address using InetAddress class

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

An IP address is represented by the Java InetAddress class. the Java.net domain.The InetAddress class offers methods for obtaining the IP address of any host name, such as www.facebook.com, www.google.com.

An unsigned integer of length 32 or 128 bits is used to represent an IP address. An IP address and its associated host name are represented by an instance of InetAddress. Addresses come in two varieties: multicast and unicast. A single interface is identified by a Unicast, while a group of interfaces is identified by a Multicast.

- IP Address:

An IP address is a numerical representation that is used to identify a particular resource on the network. The majority of networks mix TCP (Transmission Control Protocol) and IP. It creates a virtual link between the source and the destination.

- TCP/IP Protocol:

A communication protocol model called TCP/IP is used to link devices over an internet-based network. Data packet addressing, transmission, routing, and reception via the internet are made easier by TCP/IP. This communication paradigm uses two primary protocols, which are: Transmission Control Protocol, or TCP. The method for opening a communication channel over a network is provided by TCP. Additionally, it facilitates packet transmission both at the originator and recipient ends.

Internet Protocol, or IP. IP assigns an address to each connected internet node. To verify if the IP address is accurate and the message is routed correctly or not, it makes use of a gateway computer.

In addition, InetAddress features a caching system that keeps track of both successful and failed host name resolution attempts. A Java InetAddress is used by various networking classes, including Socket, ServerSocket, URL, DatagramSocket, and DatagramPacket, to encapsulate an IP address. It is employed to represent the machine's domain name in addition to numbers. IPV4 and IPV6 can both be handled by InetAddress.The InetAddress class employs the following factory methods and lacks any apparent constructors.

1. static InetAddress getLocalHost( ) throws UnknownHostException
2. static InetAddress getByName(String hostName) throws UnknownHostException
3. static InetAddress[ ] getAllByName(String hostName) throws unknownHostException

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write any four differences between IPV4 and IPV6
2. Write the use of getByName() and getAllByName() method.
3. Write the steps to assign IP address to your machine
4. Execute the following code and write the output

```
import java.io. *;
import java.net. *;
public class InetDemo
{
 public static void main (String [] args)
 {
 try
  {
  InetAddress ip=InetAddress.getByName("localhost");
  System.out.println("Host Name: "+ip.getHostName());
```

```
System.out.println("IP Address: "+ip.getHostAddress());
}
catch (Exception e) {System.out.println(e);}
}
}
```

5. Develop a program using InetAddress class to retrieve IP address of computer when hostname is entered by the user.

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 24: *Write program to demonstrate various methods of: URL class and URLConnection.

I Practical Significance

The Uniform resource locator in java allows the user to access the particular file or resource which might be stored on any local or remote machine. The URL specifies the complete path by which user can access file. It contains protocol address, port number and location of the particular resource.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO5 - Implements network programming in java.

IV Laboratory Learning Outcome(s)

LLO 24.1 Retrieve various components of URL using different methods of URL and URLConnection class

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

The URL provides easy to understand format to uniquely identify or address information on the internet. URLs are ample; every browser uses them to identify information on the Web. URL class provides a simple, concise API to access Information across the Internet using URLs.URLConnection is a class which is used to access the attributes of remote resource.

Once we made a connection to the remote resource, we are ready to use URLConnection to check the properties of remote object before actually transporting it locally. The URL and URLConnection classes are good enough for simple programs that want to connect to HTTP servers to fetch content.

- List of URL Class Methods and Constructors:

1. public URL (String protocol, String host, int port, String file) throws MalformedURLException
2. public URL (String protocol, String host, String file) throws MalformedURLException
3. public URL(String url) throws MalformedURLException
4. public URL(URL context, String url) throws MalformedURLException

- Methods:

1. public String getPath()

2. public String getAuthority()
3. public String getHost()
4. public String getFile()

- List of URLConnection class Methods and Constructors:

The openConnection() method returns a java.net.URLConnection, an abstract class whose subclasses represent the various types of URL connections.

- Methods:

1. Object getContent()
2. String getContentEncoding()
3. int getContentLength()
4. String getContentType()
5. public URL getURL()

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write the use of openConnection() method of URLConnection class.
2. Write the name of exception that can be thrown by URL class
3. Name the package in which URL class is defined.
4. Write a program using URL class to retrieve the host, protocol, port and file of URL http://www.msbte.org.in

……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 25: *Write program that demonstrates connection oriented communication using socket.

I Practical Significance

Java provides the socket programming approach for communication between the client and server. A user can write the code for both client and server as well for UDP & TCP datagram packets. By using java's network communication feature we can create interactive application to communicate within a network.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 25.1 Implement client-server TCP based communication.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

The ServerSocket class is used to create servers that listen for either local port or remote client programs to connect to them on published ports.

Constructors for ServerSocket class

1. public ServerSocket(int port) throws IOException
2. public ServerSocket(int port, int backlog) throws IOException
3. public ServerSocket(int port, int backlog, InetAddress address) throws IOException
4. public ServerSocket() throws IOException

Methods for ServerSocket class

1. public int getLocalPort()
2. public Socket accept() throws IOException
3. public void setSoTimeout(int timeout)
4. public void bind(SocketAddress host, int backlog)

The java.net.Socket class is used to communicate between client and server.The client can obtain object by creating its instance whereas the server obtains a Socket object from the return value of the accept() method.

Constructors for Socket class:

1. public Socket(String host, int port) throws UnknownHostException, IOException.
2. public Socket(InetAddress host, int port) throws IOException
3. public Socket(String host, int port, InetAddress localAddress, int localPort) throws IOException.
4. public Socket(InetAddress host, int port, InetAddress localAddress, int localPort) throws IOException.
5. public Socket()

Methods for Socket class:

1. public void connect(SocketAddress host, int timeout) throws IOException
2. public InetAddress getInetAddress()
3. public int getPort()
4. public int getLocalPort()
5. public void close() throws IOException

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write the default port of used by various services such as FTP, SMTP, HTTP.
2. Write the constructor to allow the server for waiting queue
3. Write the function of Connect(), Bind()
4. Write a program to check credentials of users (Client will send user id and password to server and server will authenticate the client using equals())
5. Write a program using Socket and ServerSocket to create Chat Application

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 26: Write program to demonstrate sending and receiving data through datagram.

I Practical Significance

The User Datagram Protocol (UDP) is connectionless, and unreliable protocol by which the user can send short messages called datagrams. The java provides DatagramSocket and DatagramPacket class to implement the concept of user datagram protocol. By using these classes, we can transfer the data asynchronous manner.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO4 - Develop java program for implementing event handling using window-based application components

IV Laboratory Learning Outcome(s)

LLO 26.1 Implement client server UDP based communication .

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

The Java DatagramSocket class is used to send and receive the datagrams it is connection less socket. That is there is no guarantee of message delivery.

It has following constructors

1. DatagramSocket() throws SocketEeption
2. DatagramSocket(int port) throws SocketEeption
3. DatagramSocket(int port, InetAddress address) throws SocketEeption

- Methods:

1. void bind(SocketAddress addr):It binds the DatagramSocket to a specific address and port.
2. void close():It closes the datagram socket.
3. void connect(InetAddress address, int port):It connects the socket to a address for the socket.
4. void disconnect():It disconnects the socket..
5. DatagramChannel getChannel():It returns the unique DatagramChannel object of datagram socket.
6. InetAddress getInetAddress():It returns the address to where the socket is connected.
7. InetAddress getLocalAddress():It gets the local address to which the socket is connected.

8. int getLocalPort():It returns the port number on the local host to which the socket is bound.
9. int getPort():It returns the port number to which the socket is connected.
10. void send(DatagramPacket p):It sends the datagram packet from the socket.
11. void receive(DatagramPacket p):It receives the datagram packet from the socket.

The Java DatagramPacket is class that can be used to send the packets. If you send multiple packet, it may arrive in any order. Additionally, packet delivery is not guaranteed.
It has following constructors
1. DatagramPacket(byte[] barr, int length)
2. DatagramPacket(byte[] barr, int length, InetAddress address, int port)

- Methods

1. InetAddress getAddress():It returns the IP address of the machine to which the datagram is being sent .
2. int getLength():It returns the length of the data to be sent or the length of the data received.
3. int getPort():It returns the port number on the remote host to which the datagram is being sent or from which the datagram was received.
4. SocketAddress getSocketAddress():It gets the SocketAddress (IP address + port number) of the remote host that the packet is being sent to or is coming from.
5. void setAddress(InetAddress iaddr):It sets the IP address of the machine to which the datagram is being sent.
6. void setLength(int length):It sets the length of the packet.
7. void setPort(int iport):It sets the port number on the remote host to which the datagram is being sent.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write the difference between SeverSocket and DatagramPacket
2. Write the steps to assign IP address to your machine
3. Write a program using DatagramPacket and DatagramSocket to create chat application
4. Write a program using DatagramPacket and DataGramSocket to copy the contents of one file into other.
5. Write a program using DatagramPacket and DatagramSocket to transfer the file from one location to another.

……
……
……
……
……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 27: *Write program to: Create sample database. Make connectivity with database.

I Practical Significance

A common application interface (API) that connects the Java programming language to a variety of databases, including Oracle, SQL, PostgreSQL, MongoDB, and others, is called Java Database Connectivity. In essence, it links the backend—which stores user-provided data in the table details—and the front end, which facilitates user interaction.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO5- Implements network programming in java

IV Laboratory Learning Outcome(s)

LLO 27.1 Make database connectivity using appropriate JDBC driver.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

JDBC API enables the application to interact with the different types of databases.It is possible to publish vital information from a remote database on a webpage using the Java applet. JDBC is a low-level API is used to invoke or call SQL command directly. The required SQL statements are passed as a 'string' to java methods

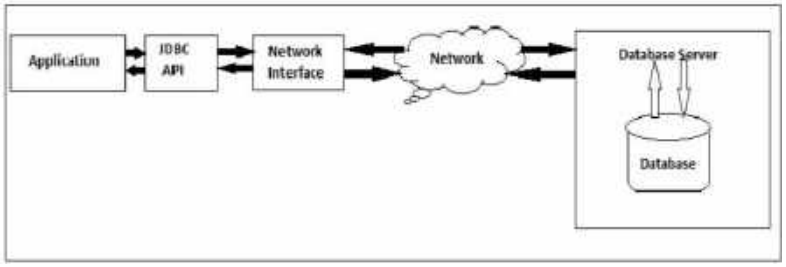


Fig no.: 27.1

Some of the current JDK add more features to JDBC that is embedded SQL which makes direct mapping of relational databases to java classes. Embedded SQL enables mixing of java into a SQL statement These statements are translated into JDBC calls using SQL processor. In this type of direct mapping, each row of the table becomes an instance of class and each column value corresponds to an attribute of that instance.

- JDBC DRIVERS:-

1. JDBC-ODBC Bridge: -.
2. Native-API: - partly java Driver: -
3. JDBC-Network pure java driver:
4. Native_protocol (100%) pure Java drivers: -

- STEPS FOR USING JDBC

There are seven steps for using JDBC to access a database.

1. Import the Package:
   - import java.sql.*;

2. Register Driver or If register then load driver using Class.forName():
   - Class.forName("com.mysql.jdbc.Driver");

3. Connect to Database:
   - Connection con = DriverManager.getConnection(
   "jdbc:mysql://localhost:0001/db", "des", "9876");

4. Create a Statement:
   - con = connection.connectDB();
     String sl = "select * from Student";
     p = con.prepareStatement(sl);

5. Execute the Statement: -
   - p.executeQuery();

6. Retrieve the Results:
   - result.next()

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a Program to create a Student Table in database and insert a record in a Student table.
2. List the advantages of JDBC over ODBC?
3. Write the Use of Class.forName()?
4. Write the steps to establish DSN oriented connection and DSNLess connection.

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 28: *Write program to implement following operations on database: Insert record, Update record, Delete record.

I Practical Significance

ODBC isn't appropriate for direct use from the Java programming language because it uses a C interface. The JDBC API was modeled after ODBC, but, because JDBC is a Java API, it offers a natural Java interface for working with SQL. JDBC is needed to provide a "pure Java" solution for application development.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO5 - Implements network programming in java.

IV Laboratory Learning Outcome(s)

LLO 28.1 Manage database using JDBC.

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

In SQL language, these fundamental operations are called INSERT, SELECT, UPDATE, and DELETE statements. The query syntax used is standard SQL, which is supported by most relational database systems, therefore even though Oracle Database is the target database system, the same techniques can be applied to other database systems as well.

Using the Java programming language, the java.sql package offers APIs for relational database data processing and access. The java.sql package contains a number of significant interfaces and classes.

Class.forName(): In this method, the runtime loads the driver's class file into memory. There is no need to create or use new objects.

- Class.forName("oracle.jdbc.driver.OracleDriver");

int executeUpdate(String sql): INSERT, UPDATE, or DELETE statements are executed by int executeUpdate(String sql), which returns a changed account with the number of rows affected.

- Connectioncon=DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1011:orcl", "Page1", "page2");

- Statement stmt = con.createStatement();

1.INSERT:

```
String q1 = "insert into userid values('" +id+ "',  '" +name+ "', '" +email+ "')";
        int a = stmt.executeUpdate(q1);
```

2.UPDATE:

```
String q1 = "UPDATE userid set name = '" + newname + "' WHERE id = '" +id+ "'      AND
email = '" +email+ "'";
        int a = stmt.executeUpdate(q1);
```

3.DELETE

```
String q1 = "DELETE from userid WHERE id = '" + id + "' AND name = '" + name + "'";
int a = stmt.executeUpdate(q1);
```

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……

……

……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is INSERT, DELETE and UPDATE operation in DBMS?
2. How do you UPDATE multiple records in a database?
3. Write a program for Implementing Insert Statement.
4. Write a program for Implementing Update Statement.
5. Write a program for Implementing Delete Statement.

……

……

……

……

……

……

……

……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 29: Write program to demonstrate the use of PreparedStatement.

I Practical Significance

A Java JDBC PreparedStatement is a special kind of Java JDBC Statement object with some useful additional features. Remember, we need a Statement in order to execute either a query or an update. We can use a Java JDBC PreparedStatement instead of a Statement and benefit from the features of the PreparedStatement.

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO5 - Implements network programming in java

IV Laboratory Learning Outcome(s)

LLO 29.1 Manage database using JDBC

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

Prepared Statement Interface:

PreparedStatement helps us in preventing SQL injection attacks because it automatically escapes the special characters. PreparedStatement allows us to execute dynamic queries with parameter inputs. PreparedStatement provides different types of setter methods to set the input parameters for the query. PreparedStatement is faster than Statement. It becomes more visible when we reuse the PreparedStatement or use it's batch processing methods for executing multiple queries. PreparedStatement helps us in writing object Oriented code with setter methods whereas with Statement we have to use String Concatenation to create the query. If there are multiple parameters to set, writing Query using String concatenation looks very ugly and error prone. PreparedStatement returns FORWARD_ONLY ResultSet, so we can only move in forward direction. Unlike Java Arrays or List, the indexing of PreparedStatement variables start with 1.

One of the limitations of PreparedStatement is that we can't use it for SQL queries with IN clause because PreparedStatement doesn't allow us to bind multiple values for single placeholder (?).

Methods of Prepared Statements.

1. public void setInt(int paramIndex, int value):-sets the integer value to the given parameter index.

2. public void setString(int paramIndex, String value) :-sets the String value to the given parameter index.

3. public void setFloat(int paramIndex, float value):- sets the float value to the given parameter index.

4. public void setDouble(int paramIndex, double value):-sets the double value to the given parameter index.

5. public int executeUpdate() :-executes the query. It is used for create, drop, insert, update, delete etc.

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain Advantages of Prepared Statement Interface.
2. Explain disadvantages of Prepared Statements
3. Develop JDBC program to retrieve data.
4. Develop a program to update a record in database table

……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	

Practical No. 30: *Write program to retrieve data from table using ResultSet interface. (Use various methods of navigation methods)

I Practical Significance

A single row of the query results is pointed to by the cursor/pointer that ResultSet keeps. We can access database records one by one by iterating through them and using the getter and navigation methods that ResultSet provides. Data updates are also possible with ResultSet

II Industry / Employer Expected Outcome(s)

Develop standalone and network-based applications using Java.

III Course Level Learning Outcomes(s)

CO5 - Implements network programming in java

IV Laboratory Learning Outcome(s)

LLO 30.1 Implement dynamic query

V Relevant Affective Domain related Outcomes

1. Follow precautionary measures.
2. Follow naming conventions.
3. Follow ethical practices

VI Relevant Theoretical Background

ResultSet Interface:

The SQL statements that read data from a database query, return the data in a result set. The SELECT statement is the standard way to select rows from a database and view them in a result set. The java.sql.ResultSet interface represents the result set of a database query.

A ResultSet object maintains a cursor that points to the current row in the result set.

The term "result set" refers to the row and column data contained in a ResultSet object.

The methods of the ResultSet interface can be broken down into three categories −

- Navigational methods: Used to move the cursor around.
- Get methods: Used to view the data in the columns of the current row being pointed by the cursor.
- Update methods: Used to update the data in the columns of the current row. The updates can then be updated in the underlying database as well. It is movable based on the properties of the ResultSet. These properties are designated when the corresponding Statement The cursor that generates the ResultSet is created.

JDBC provides the following connection methods to create statements with desired ResultSet −

- createStatement(int RSType, int RSConcurrency);
- prepareStatement(String SQL, int RSType, int RSConcurrency);
- prepareCall(String sql, int RSType, int RSConcurrency);

The methods of the ResultSet interface can be broken down into three categories −

- Navigational methods: Used to move the cursor around.
- Get methods: Used to view the data in the columns of the current row being pointed by the cursor.
- Update methods: Used to update the data in the columns of the current row. The updates can then be updated in the underlying database as well. JDBC provides the following connection methods to create statements with desired ResultSet −
- createStatement(int RSType, int RSConcurrency);
- prepareStatement(String SQL, int RSType, int RSConcurrency);
- prepareCall(String sql, int RSType, int RSConcurrency);

The first argument indicates the type of a ResultSet object and the second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable

VII Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				

VIII Conclusion

……
……
……

IX Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain the methods of ResultSet Interface.
2. Explain Types of ResultSet
3. Write program to retrieve data from table using ResultSet interface. (Use various methods of navigation methods)

……
……
……
……
……
……
……
……
……
……

X References:

1. https://www.javatpoint.com
2. https://www.tutuorialspoint.com

XI Assessment Scheme (50 Marks)

S. No.	Weightage- Process related: 60%	Marks-35
1.	Logic formation:30%	
2.	Debugging ability:20%	
3.	Follow ethical practices:10%	
	Weightage- Product related: 40%	**Marks-15**
4.	Expected output:15%	
5.	Timely Submission:15%	
6.	Answer to sample questions:10%	
	Total 50	
	Dated Signature of Course Teacher	