BAPATLA WOMEN'S ENGINEERING COLLEGE

JAVA PROGRAMMING LAB/EC 264

II/IV B.TECH – II SEM-R22

LABORATORY MANUAL

PREPARED BY

MS. M. BHAVANI M.Tech

ASSISTANT PROFESSOR

DEPARTMENT OF E.C.E

&

MRS.B.MAHA LAKSHMI M.Tech.(Ph.D)

ASSISTANT PROFESSOR

DEPARTMENT OF E.C.E

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

(AFFILIATED TO ACHARYA NAGARJUNA UNIVERSITY)

BAPATLA, BAPATLA DISTRICT,ANDHRA PRADESH,INDIA PIN:522101

(A.Y:2023-24)

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

AN ISO 9001-2015 CERTIFIED INSTITUTION

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE VISION AND MISSION

Vision

Bapatla Women's Engineering College has been established with forethought to organize the growing needs of higher technical education for WOMEN, in order to provide them a competative edge and a unique niche in the society.

Mission

Our mission is to impart the quality education at par with global standards to the students from all over India and in particular those from the local and rural areas. We continuously try to maintain high standards so as to make them technologically competent and ethically strong individuals who shall be able to improve the quality of life and economy of our country.

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

AN ISO 9001-2015 CERTIFIED INSTITUTION

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION

To produce highly professional and socially responsible women engineers through world class quality technical education and research to meet industry and Society needs in the field of Electronics and Communication Engineering.

MISSION

M1- To inculcate quality education for women through innovative learning practices with the use of modern tools.

M2- To impart professional ethics and moral values through co-curricular and extra-curricular activities as individual and team member.

M3- Empowering graduates through industrial trainings/on-job training in collaboration with industries with-a-view to life-long learning.

Department Programme Educational Objectives (PEOs)

PEO1: To develop and implementation of Electronics and Communication systems by imparting technical knowledge, professional leadership and communication skills as an individual and team in a multi-disciplinary stream.

PEO2: To train Graduates in solving complex problems, analyzing, designing and imparting research-based knowledge by using modern scientific tools to develop least cost-effective and innovative products.

PEO3: To produce women graduates through life-long learning, who ensure ethical and moral behavior.

The Programme Specific Objectives(PSO'S)

PSO1: To understand and analyze basic concepts of electronics and communication in various areas like signal processing, VLSI, embedded systems, communications, digital and analog devices.

PSO2: Ability to design and develop computing systems using concepts of mathematics, electronics engineering and other related inter disciplines to asses and resolve societal, environmental, health safety and cultural issues with professional ethics.

Department Programme Outcomes (POs)

- **PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **PO2: Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools prediction and modelling to complex engineering activities with an understanding of the limitations.
- **PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- **PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
- **PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering prac tice.
- **PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and receive clear instructions.
- **PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in team, to manage projects and in multidisciplinary environments.
- **PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

EC264-Java Programming Lab **2023-2024**

SYLLABUS

EC264-Java Programming Lab

L	T	P	M	C
0	0	3	100	2

OBJECTIVES:

- ✓ To prepare students to become familiar with the Standard Java technologies of J2SE
- ✓ To prepare students to excel in Object Oriented programming and to succeed as a Java Developer through global rigorous education.
- ✓ To provide Students with a solid foundation in OOP fundamentals required to solve programming problems and also to learn Advanced Java topics like J2ME, J2EE, JSP, JavaScript
- ✓ To train Students with good OOP programming breadth so as to comprehend, analyze, design and create novel products and solutions for the real life problems.
- ✓ To inculcate in students professional and ethical attitude, multidisciplinary approach and an ability to relate java programming issues to broader application context.
- ✓ To provide student with an academic environment aware of excellence, written ethical codes and guidelines and lifelong learning needed for a successful professional career.

LIST OF EXPERIMENTS

Basic Programs:

1. Write java programs to find the following
   a) largest of given three numbers
   b) reverses the digits of a number
   c) given number is prime or not
   d) GCD of given two integers
2. Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
3. Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.
4. The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions.

Matrices, Overloading, Overriding:

5. Write a java program to multiply two given matrices.
6. Write a java program to implement method overloading and constructors overloading.

7. Write a java program to implement method overriding.

Palindrome, Abstract Class:

8. Write a java program to check whether a given string is palindrome.
9. Write a java program for sorting a given list of names in ascending order.
10. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape

Interface:

11. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Multithreading:

12. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
13. Write a java program that correct implements of producer consumer program
14. Write a program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

Files:

15. Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
16. Write a java program that displays the number of characters, lines and words in a text file.

17. Write a java program that reads a file and displays the file on the screen with line number before each line.

Experiment(s) beyond syllabus

1. Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

OUTCOMES:

Upon successful completion of this course, the students will be able to:

- ✓ Able to analyze the necessity for Object Oriented Programming paradigm and over structured programming and become familiar with the fundamental concepts in OOP.
- ✓ Demonstrate an ability to design and develop java programs, analyze, and interpret object oriented data and report results.
- ✓ Demonstrate an ability to design an object oriented system, AWT components or multithreaded process as per needs and specifications.
- ✓ Demonstrate an ability to visualize and work on laboratory and multidisciplinary tasks like console and windows applications both for standalone and Applets programs

Web References:

1. http://java.sun.com
2. http://www.oracle.com/technetwork/java/index.html
3. http://java.sun.com/javase

EC264 –Java Programming Lab **2023-2024**

LIST OF EXPERIMENTS

Basic Programs:

1. Write java programs to find the following
   a) largest of given three numbers
   b) reverses the digits of a number
   c) given number is prime or not
   d) GCD of given two integers
2. Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
3. Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.
4. The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions.

Matrices, Overloading, Overriding:

5. Write a java program to multiply two given matrices.
6. Write a java program to implement method overloading and constructors overloading.
7. Write a java program to implement method overriding.

Palindrome, Abstract Class:

8. Write a java program to check whether a given string is palindrome.
9. Write a java program for sorting a given list of names in ascending order.
10. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape

Interface:

11. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were

not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Multithreading:

12. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
13. Write a java program that correct implements of producer consumer program
14. Write a program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

Files:

15. Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
16. Write a java program that displays the number of characters, lines and words in a text file.
17. Write a java program that reads a file and displays the file on the screen with line number before each line.

Experiment(s) beyond syllabus

1. Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

AN ISO 9001-2015 CERTIFIED INSTITUTION

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Course Outcomes (COs):

After completion of this course, students will be able to –

CO1	Develop and implement Java programs for simple applications that make use of classes
CO2	Design applications using file processing
CO3	Apply the concepts of classes, packages, interfaces, exception handling
CO4	Develop and implement Java programs with arraylist
CO5	Build software development skills using java programming for real-world applications

Mapping of Course Outcomes with PSOs and POs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	-	-	-	-	-	-	-	-	2	2
CO2	3	2	2	1	-	-	-	-	-	-	-	-	2	2
CO3	3	2	2	1	-	-	-	-	-	-	-	-	2	2
CO4	3	2	2	1	-	-	-	-	-	-	-	-	2	2
CO5	3	2	2	1	-	-	-	-	-	-	-	-	3	2
AVG_CO	3	2	2	1	-	-	-	-	-	-	-	-	2	2

Correlation Levels:---1: Low 2: Medium 3: Strong

EC264–Java Programming Lab **2023-2024**

INDEX

Expt. No.	Name of the Experiment	Page No.
	INTRODUCTION TO JAVA PROGRAMMING	**13-35**
	BASIC PROGRAMS:	**25-29**
1	Write java programs to find the following a) largest of given three numbers b) reverses the digits of a number c) given number is prime or not d) GCD of given two integers	25-29
2	Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.	30-31
3	Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.	32-33
4	The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions.	34-35
	MATRICES, OVERLOADING, OVERRIDING:	**36-40**
5	Write a java program to multiply two given matrices.	36-37
6	Write a java program to implement method overloading and constructors overloading.	38-39
7	Write a java program to implement method overriding.	40
	PALINDROME, ABSTRACT CLASS:	**41-46**
8	Write a java program to check whether a given string is palindrome.	41-42
9	Write a java program for sorting a given list of names in ascending order.	43-44
10	Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape	45-46

	INTERFACE	**47-49**
11	Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.	47-49
	MULTITHREADING:	**50-60**
12	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.	50-52
13	Write a java program that correct implements of producer consumer Program	53-56
14	Write a program that creates three threads. First thread displays "GoodMorning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.	57-60
	FILES:	**61-66**
15	Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.	61-62
16	Write a java program that displays the number of characters, lines and words in a text file.	63-64
17	Write a java program that reads a file and displays the file on the screen with line number before each line.	65-66
	Experiment(s) beyond syllabus	
1	Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.	67-68

INTRODUCTION TO JAVA PROGRAMMING

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

Application

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc.

FEATURES OF JAVA

Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, Java language is a simple programming language because:

- Java syntax is based on C++ (so easier for programmers to learn it after C++).
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporate both data and behaviour.

Platform Independent

Java is platform independent because it is different from other languages like C, C++, etc. which

are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

Secured

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

- No explicit pointer
- Java Programs run inside a virtual machine sandbox

Robust

The English mining of Robust is strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.

Architecture-natural

Java is architecture natural because there are no implementation dependent features, for example, the size of primitive types is fixed.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code.

Distributed

Java is distributed because it facilitates users to create distributed applications in Java.

Multi-threaded

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads.

Dynamic

- Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

JAVA PROGRAM:

The requirement and structure for Java Hello World Example

For executing any Java program, the following software or application must be properly installed

- Install the JDK if you don't have installed it, download the JDK and install it.
- Set path of the jdk/bin directory. http://www.javatpoint.com/how-to-set-path-in-java
- Create the Java program.
- Compile and run the Java program

Creating Hello World Example

```
class Simple{
  public static void main(String args[]){
   System.out.println("Hello Java");
  }
}
```

File Name: Simple.java.

Compile: javac Simple.java

Run: java Simple

Output: Hello Java

1. Comments: Comments are used for explaining code and are used in a similar manner in Java or C or C++. Compilers ignore the comment entries and do not execute them. Comments can be of a single line or multiple lines.

Single line Comments:

Syntax:

```
// Single line comment
```

Multi-line **comments:**

Syntax:

```
/* Multi line comments*/
```

2. import java.io.*: This means all the classes of the io package can be imported. Java io package provides a set of input and output streams for reading and writing data to files or other input or output sources.

3. class: The class contains the data and methods to be used in the program. Methods define the behavior of the class. Class **GFG** has only one method Main in JAVA.

4. static void main(): static keyword tells us that this method is accessible without instantiating the class.

5. void: keywords tell that this method will not return anything. The main**()** method is the entry point of our application.

6. System.in: This is the **standard input stream** that is used to read characters from the keyboard or any other standard input device.

7. System.out: This is the **standard output stream** that is used to produce the result of a program on an output device like the computer screen.

8. println(): This method in Java is also used to display text on the console. It prints the text on the console and the cursor moves to the start of the next line at the console. The next printing takes place from the next line.

9. String []args: This is the argument passed to the main function which is an array of strings with the array name args. One can choose their own flexible name but this name is used by many developers.

Everything in java is represented in Class as an object including the main function.

HOW TO SET PATH IN JAVA

The path is required to be set for using tools such as javac, java, etc.

If you are saving the Java source file inside the JDK/bin directory, the path is not required to be set because all the tools will be available in the current directory.

However, if you have your Java file outside the JDK/bin folder, it is necessary to set the path of JDK.

There are two ways to set the path in Java:

1. Temporary
2. Permanent

1) How to set the Temporary Path of JDK in Windows

To set the temporary path of JDK, you need to follow the following steps:

- Open the command prompt
- Copy the path of the JDK/bin directory
- Write in command prompt: set path=copied_path

For example:set path=C:\Program Files\Java\jdk1.7.0_23\bin

How to set Permanent Path of JDK in Windows

For setting the permanent path of JDK, you need to follow these steps:

Go to MyComputer properties -> advanced tab -> environment variables -> new tab of user variable -> write path in variable name -> write path of bin folder in variable value -> ok -> ok -> ok

1) Go to MyComputer properties
2) Click on the advanced tab
3) Click on environment variables
4) Click on the new tab of user variables
5) Write the path in the variable name
6) Copy the path of bin folder
7) Paste path of bin folder in the variable value
8) Click on ok button
9) Click on ok button

Now your permanent path is set. You can now execute any program of java from any drive.

DIFFERENCE BETWEEN JDK, JRE, AND JVM

JVM

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

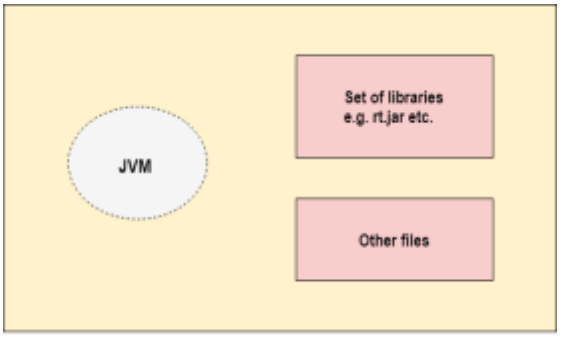
JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. There are three notions of the JVM: *specification*, *implementation*, and *instance*.

The JVM performs the following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JRE

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.The implementation of JVM is also actively released by other companies besides Sun Micro Systems.



JRE

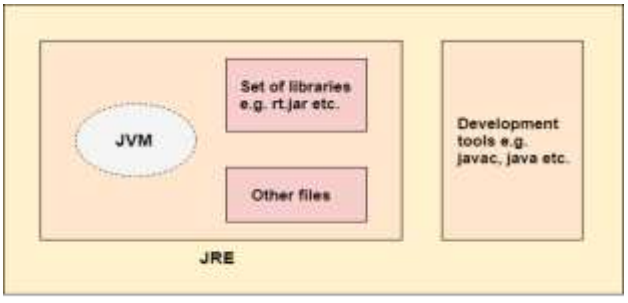
JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.



JDK

JAVA CONTROL STATEMENTS

Java provides statements that can be used to control the flow of Java code. Such statements are called control flow statements

Java provides three types of control flow statements.

1. Decision Making statements
    - if statements
    - switch statement
2. Loop statements
    - do while loop
    - while loop
    - for loop
    - for-each loop
3. Jump statements
    - break statement
    - continue statement

Decision-Making statements:

As the name suggests, decision-making statements decide which statement to execute and when. Decision-making statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided. There are two types of decision-making statements in Java, i.e., If statement and switch statement.

1) If Statement:

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. In Java, there are four types of if-statements given below.

1. Simple if statement
2. if-else statement
3. if-else-if ladder
4. Nested if-statement

1)Simple if statement:

It is the most basic statement among all control flow statements in Java. It evaluates a Boolean expression and enables the program to enter a block of code if the expression evaluates to true.

Syntax

```
if(condition) {
statement 1; //executes when condition is true
}
```

2) if-else statement

The if-else statement is an extension to the if-statement, which uses another block of code, i.e., else block. The else block is executed if the condition of the if-block is evaluated as false.

Syntax

```
if(condition) {
statement 1; //executes when condition is true
}
else{
statement 2; //executes when condition is false
}
```

3) if-else-if ladder:

The if-else-if statement contains the if-statement followed by multiple else-if statements. In other words, we can say that it is the chain of if-else statements that create a decision tree where the program may enter in the block of code where the condition is true. We can also define an else statement at the end of the chain.

Syntax of if-else-if statement is given below.

```
if(condition 1) {
statement 1; //executes when condition 1 is true
}
else if(condition 2) {
statement 2; //executes when condition 2 is true
}
```

```
else {
statement 2; //executes when all the conditions are false
}
```

4. Nested if-statement

In nested if-statements, the if statement can contain a **if** or **if-else** statement inside another if or else-if statement.

Syntax of Nested if-statement is given below.

```
if(condition 1) {
statement 1; //executes when condition 1 is true
if(condition 2) {
statement 2; //executes when condition 2 is true
}
else{
statement 2; //executes when condition 2 is false
}
}
```

Switch Statement:

In Java, Switch statements are similar to if-else-if statements. The switch statement contains multiple blocks of code called cases and a single case is executed based on the variable which is being switched. The switch statement is easier to use instead of if-else-if statements. It also enhances the readability of the program.

Points to be noted about switch statement:

- The case variables can be int, short, byte, char, or enumeration. String type is also supported since version 7 of Java
- Cases cannot be duplicate
- Default statement is executed when any of the case doesn't match the value of expression. It is optional.

- Break statement terminates the switch block when the condition is satisfied.
  It is optional, if not used, next case is executed.
- While using switch statements, we must notice that the case expression will be of the same type as the variable. However, it will also be a constant value.

The syntax to use the switch statement is given below.

```
switch (expression){
   case value1:
   statement1;
   break;
   .
   .
   .
   case valueN:
   statementN;
   break;
   default:
    default statement;
}
```

Loop Statements

In programming, sometimes we need to execute the block of code repeatedly while some condition evaluates to true. However, loop statements are used to execute the set of instructions in a repeated order. The execution of the set of instructions depends upon a particular condition

In Java, we have three types of loops that execute similarly. However, there are differences in their syntax and condition checking time.

1. for loop
2. while loop
3. do-while loop

Let's understand the loop statements one by one.

Java for loop

In Java, for loop is similar to C and C++. It enables us to initialize the loop variable, check the condition, and increment/decrement in a single line of code. We use the for loop only when we exactly know the number of times, we want to execute the block of code.

```
for(initialization, condition, increment/decrement) {
//block of statements
}
```

Java for-each loop

Java provides an enhanced for loop to traverse the data structures like array or collection. In the for-each loop, we don't need to update the loop variable. The syntax to use the for-each loop in java is given below.

```
for(data_type var : array_name/collection_name){
//statements
}
```

Java while loop

The while loop is also used to iterate over the number of statements multiple times. However, if we don't know the number of iterations in advance, it is recommended to use a while loop. Unlike for loop, the initialization and increment/decrement doesn't take place inside the loop statement in while loop.

It is also known as the entry-controlled loop since the condition is checked at the start of the loop. If the condition is true, then the loop body will be executed; otherwise, the statements after the loop will be executed.

The syntax of the while loop is given below.

```
while(condition){
//looping statements
}
```

Jump Statements

Jump statements are used to transfer the control of the program to the specific statements. In other words, jump statements transfer the execution control to the other part of the program. There are two types of jump statements in Java, i.e., break and continue.

Java break statement

As the name suggests, the break statement is used to break the current flow of the program and transfer the control to the next statement outside a loop or switch statement. However, it breaks only the inner loop in the case of the nested loop.
The break statement cannot be used independently in the Java program, i.e., it can only be written inside the loop or switch statement.

Java continue statement

Unlike break statement, the continue statement doesn't break the loop, whereas, it skips the specific part of the loop and jumps to the next iteration of the loop immediately.

PROGRAM-1

1. **Question: Write java programs to find the following**
   - **a) largest of given three numbers**
   - **b) reverses the digits of a number**
   - **c) given number is prime or not**
   - **d) GCD of given two integers**

1. (a) largest of given three numbers

Aim: To write a java program to find largest of given three numbers.

Source Code:

```
import java.util.Scanner;
public class largestnum
{
public static void main(String args[])
{
int a, b, c;
System.out.println("Enter three integers: ");
Scanner s1 = new Scanner(System.in);
a=s1.nextInt();
b=s1.nextInt();
c=s1.nextInt();
if (a > b && a > c)
System.out.println("The largest number is: "+a);
else if (b > a && b > c)
System.out.println("The largest number is: "+b);
else if (c > a && c > b)
System.out.println("The largest number is: "+c);
else
System.out.println("The numbers are same.");
}
}
```

File Name: largestnum.java

Compile:javac largestnum.java

Run:java largestnum

Output:

Enter three integers:

34

78

12

The largest number is: 78

Enter three integers:

3

3

3

The numbers are same.

1.(b) reverses the digits of a number

Aim: To write a java program to reverses the digits of a number

Source Code:

```
import java.util.*;
public class ReverseNumber
{
public static void main(String args[])
{
int number, reverse = 0,remainder=0;
Scanner s1=new Scanner(System.in);
System.out.println("enter a number");
number=s1.nextInt();
while (number>0)
{
remainder = number % 10;
reverse = (reverse * 10) + remainder;
number=number/10;
}
System.out.println("The reverse of the given number is: " + reverse);
}
}
```

Output:

enter a number

6789023

The reverse of the given number is: 3209876

1. (c) given number is prime or not

Aim: To write a java program to check whether the given number is prime or not

Source Code:

```
import java.util.Scanner;
class Prime
{
        public static void main(String arg[])
        {

          System.out.println("Enter a number ");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        primeCal(n);
        }
  static void primeCal(int num)
  {
        int count=0;
        for(int i=1;i<=num;i++)
        {
          if(num%i==0)
          {
              count++;
          }
        }
        if(count==2)
             System.out.println("prime number ");
        else
        System.out.println("Not a prime number ");
  }
}
```

Output:

Enter a number

5

prime number

Enter a number

6

Not a prime number

1. (d) GCD of given two integers

Aim: To write a java program to find GCD of given two integers

Source Code:

```
import java.util.Scanner;
public class GCD
{
public static void main(String args[])
{
int a, b, gcd = 0;
Scanner s1 = new Scanner(System.in);
System.out.print("Enter the First Number: ");
a = s1.nextInt();
System.out.print("Enter the Second Number: ");
b = s1.nextInt();
gcd = findGCD(a, b);
System.out.println("GCD of " + a + " and " + b + " = " + gcd);
}
public static int findGCD(int a, int b)
{
while(b != 0)
{
if(a > b)
{
a = a - b;
}
else
{
```

```
b = b - a;
}
}
return a;
}
}
```

Output:

```
Enter the First Number: 23
Enter the Second Number: 67
GCD of 23 and 67 =  1
Enter the First Number: 12
Enter the Second Number: 20
GCD of 12 and 20 =  4
```

VIVA QUESTIONS:

1. What do you know about Java?
2. Why the main method is static in java?
3. List any five features of Java?
4. What method did you use to find the GCD of two numbers?
5. List any two Java IDE's?

PROGRAM-2

2. Try debug step by step with small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

Aim: To write a java program to find factorial of a program using for loop and if else condition.

Source Code:

```
import java.util.Scanner;
public class factorialpro
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the number :");
      int num = sc.nextInt();
      if(num>=0)
      {
      int fact=1;
      for(int i=1;i<=num;i++)
      {
         fact=fact*i;
      }
        System.out.println("The factorial of the entered number is :"+fact);
      }
      else
      {
         System.out.println("Factorial not possible.");
         System.out.println("Please enter valid input.");
      }
   }

}
```

Output:

Enter the number :

3

The factorial of the entered number is :6

D:\JAVA\bin>java factorialpro

Enter the number :

9

The factorial of the entered number is :362880

D:\JAVA\bin>java factorialpro

Enter the number :

0

The factorial of the entered number is :1

VIVA QUESTIONS:

1. Explain the purpose of the if-else condition?
2. Describe the purpose of the for loop?
3. Define factorial?
4. What is the role of the Scanner class?

PROGRAM-3

3. Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.

Aim: To write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$.

Source Code:

```
import java.util.Scanner;
public class RootsOfQuadraticEquation {
  public static void main(String args[]){
    double secondRoot = 0, firstRoot = 0;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the value of a ::");
    double a = sc.nextDouble();

    System.out.println("Enter the value of b ::");
    double b = sc.nextDouble();

    System.out.println("Enter the value of c ::");
    double c = sc.nextDouble();

    double determinant = (b*b)-(4*a*c);
    double sqrt = Math.sqrt(determinant);

    if(determinant>0){
      firstRoot = (-b + sqrt)/(2*a);
      secondRoot = (-b - sqrt)/(2*a);
      System.out.println("Roots are :: "+ firstRoot +" and "+secondRoot);
    }else if(determinant == 0){
      System.out.println("Root is :: "+(-b + sqrt)/(2*a));
    }
  }
}
```

Output:

Enter the value of a ::

2

Enter the value of b ::

3

Enter the value of c ::

1

Roots are :: -0.5 and -1.0

Enter the value of a ::

15

Enter the value of b ::

68

Enter the value of c ::

3

Roots are :: -0.044555558333472335 and -4.488777774999861

Enter the value of a ::

1

Enter the value of b ::

5

Enter the value of c ::

1

Roots are :: -0.20871215252208009 and -4.7912878474779195

VIVA QUESTIONS:

1. What is a quadratic equation?
2. How do you calculate the square root of a number in Java?
3. What does the main method signify in a Java program?
4. What does Math.sqrt(determinant) do?

PROGRAM-4

4. The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions.

Aim: To Write a java program that uses both recursive and non-recursive functions.

Source Code:

Recursive

```
public class FibonacciCalc {
  public static int fibRecursion(int count) {
    if (count == 0) {
      return 0;
    }
    if (count == 1 || count == 2) {
      return 1;
    }
    return fibRecursion(count - 1) + fibRecursion(count - 2);
  }
  public static void main(String args[]) {
      int fib_len=5;
    System.out.print("Fibonacci Series of " + fib_len + " numbers is: \n");
    for (int i = 0; i < fib_len; i++) {
      System.out.print(fibRecursion(i) + " ");
    }
  }
}
```

Output:

```
Fibonacci Series of 5 numbers is:
0 1 1 2 3
Fibonacci Series of 9 numbers is:
0 1 1 2 3 5 8 13 21
```

Non-recursive

```
class Fibonaccinonrecursive{
public static void main(String args[])
{
 int n1=0,n2=1,n3,i,count=5;
 System.out.print(n1+" "+n2);
 for(i=2;i<count;++i)
 {
  n3=n1+n2;
  System.out.print(" "+n3);
  n1=n2;
  n2=n3;
 }
}
}
```

Output:

Fibonacci Series of 5 numbers is:

0 1 1 2 3

VIVA QUESTIONS:

1. What is the Fibonacci sequence?
2. What is recursive?
3. What is non- recursive?
4. Compare recursive and non- recursive?
5. What are the advantages and disadvantages of using the recursive approach for Fibonacci numbers?

PROGRAM-5

5. Write a java program to multiply two given matrices.

Aim: To write a java program to multiply two given matrices.

Source Code:

```
import java.util.Scanner;
public class MatrixMultiplication {
  public static void main(String args[]) {
  int row1, col1, row2, col2;
    Scanner s = new Scanner(System.in);
    System.out.print("Enter number of rows in first matrix: ");
    row1 = s.nextInt();
    System.out.print("Enter number of columns in first matrix: ");
    col1 = s.nextInt();
    System.out.print("Enter number of rows in second matrix: ");
    row2 = s.nextInt();
    System.out.print("Enter number of columns in second matrix: ");
    col2 = s.nextInt();
    if (col1 != row2) {
      System.out.println("Matrix multiplication is not possible");
      return;
    }
    int a[][] = new int[row1][col1];
    int b[][] = new int[row2][col2];
    int c[][] = new int[row1][col2];
    System.out.println("\nEnter values for matrix A : ");
    for (int i = 0; i < row1; i++) {
      for (int j = 0; j < col1; j++) a[i][j] = s.nextInt();
    }
    System.out.println("\nEnter values for matrix B : ");
    for (int i = 0; i < row2; i++) {
      for (int j = 0; j < col2; j++) b[i][j] = s.nextInt();
    }
    System.out.println("\nMatrix multiplication is : ");
    for (int i = 0; i < row1; i++) {
```

```
      for (int j = 0; j < col2; j++) {
        c[i][j] = 0;
        for (int k = 0; k < col1; k++) {
          c[i][j] += a[i][k] * b[k][j];
        }
        System.out.print(c[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

Output:

Enter number of rows in first matrix: 3

Enter number of columns in first matrix: 3

Enter number of rows in second matrix: 3

Enter number of columns in second matrix: 3

Enter values for matrix A :

5 2 3

2 6 3

6 9 1

Enter values for matrix B :

2 7 5

1 4 3

1 2 1

Matrix multiplication is :

15 49 34

13 44 31

22 0 58

VIVA QUESTIONS:

1. What is the purpose of the System class?
2. Explain the purpose of the nested loops in the program.
3. What are the inputs required from the user before matrix multiplication can occur?
4. What is the use of this keyword?

PROGRAM-6

6. Write a java program to implement method overloading and constructors overloading.

Aim: To write a java program to implement method overloading and constructors overloading

Source Code:

method overloading

```
class Method_Overloading {
   double figure(double l, int b)
   {
      return (l*b);
   }
   float figure(int s)
   {
      return (s*s);
   }
   public static void main(String[] args) {
      Method_Overloading obj = new Method_Overloading();
      System.out.println("Area of Rectangle: " +obj.figure(5.55, 6));
      System.out.println("Area of Square: " +obj.figure(3));
   }
}
```

Output:

```
Area of Rectangle: 33.3
Area of Square: 9.0
```

Constructors overloading

```
class Box {
   double width, height, depth;
   Box(double w, double h, double d)
   {
      width = w;
      height = h;
      depth = d;
   }
   Box() { width = height = depth = 0; }
```

```
    Box(double len) { width = height = depth = len; }
    double volume() { return width * height * depth; }
}
 public class constructerOverload {
    public static void main(String args[])
    {
        Box mybox1 = new Box(10, 20, 15);
        Box mybox2 = new Box(10);
        Box mycube = new Box(7);
        double vol;
        vol = mybox1.volume();
        System.out.println("Volume of mybox1 is " + vol);
        vol = mybox2.volume();
        System.out.println("Volume of mybox2 is " + vol);
        vol = mycube.volume();
        System.out.println("Volume of mycube is " + vol);
    }
}
```

Output:

```
D:\JAVA\bin>java constructerOverload
Volume of mybox1 is 3000.0
Volume of mybox2 is 1000.0
Volume of mycube is 343.0
```

VIVA QUESTIONS:

1. What is method overloading in Java?
2. What is constructor overloading in Java?
3. Explain the difference between constructor overloading and method overloading.
4. What are the advantages of method overloading?
5. What are the advantages of constructor overloading?

PROGRAM-7

7. Write a java program to implement method overriding.

Aim: To Write a java program to implement method overriding.

Source Code:

```
class college {
 public void move() {
  System.out.println("College is open");
 }
}
class univ extends college {
 public void move() {
  System.out.println("University is open too");
 }
}
public class studoverriding {
 public static void main(String args[]) {
  college a = new college();
  college b = new univ();
  a.move();
  b.move();
 }
}
```

Output:

College is open

University is open too

VIVA QUESTIONS:

1. What is method overriding in Java?
2. What are the advantages of method overriding?
3. Explain the difference between method overriding and method overloading.
4. Why is method overriding useful in object-oriented programming?

PROGRAM-8

8. Write a java program to check whether a given string is palindrome.

Aim: To Write a java program to check whether a given string is palindrome or not.

Source Code:

Palindrome to check number:

```
import java.util.*;
class palindrome {
  public static void main(String args[])
{
    int rem,n,rev,m;
    Scanner s1= new Scanner(System.in);
    System.out.print("Enter the number: ");
    n= s1.nextInt();
    m=n;
     for(rev=0, n=0, n=n/10)
{
      rem= n % 10;
      rev = (rev * 10) + rem;
    }
    if (m == rev){
      System.out.println("Yes, it is palindrome");
    else
      System.out.println("No, it is not palindrome");
    }
```

Output:

```
Enter the number: 121
Yes, it is palindrome
Enter the number: 345
No, it is not palindrome
```

Palindrome to check String:

Source Code:

```
class ChkPalindrome
{
  public static void main(String args[])
  {
    String str, rev = "";
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a string:");
    str = sc.nextLine();
     int length = str.length();
     for ( int i = length - 1; i >= 0; i-- )
       rev = rev + str.charAt(i);
     if (str.equals(rev))
       System.out.println(str+" is a palindrome");
    else
       System.out.println(str+" is not a palindrome");
   }
}
```

Output:

```
D:\JAVA\bin>java ChkPalindrome
Enter a string:
radar
radar is a palindrome
D:\JAVA\bin>java ChkPalindrome
Enter a string:
java
java is not a palindrome
```

VIVA QUESTIONS:

1. What is a palindrome?
2. What is the purpose of the variable rev in the program?
3. What is the use of modulus operator (%) in the loop?
4. Why main() method is public, static and void in java ?
5. Explain about main() method in java ?

PROGRAM-9

9. Write a java program for sorting a given list of names in ascending order.

Aim: To Write a java program for sorting a given list of names in ascending order.

Source Code:

```
import java.util.*;
class Sorting
{
	void sortStrings()
	{
		Scanner s = new Scanner(System.in);
		System.out.println("Enter the value of n: ");
		int n = s.nextInt();
		String[] str = new String[n];
		System.out.println("Enter strings: ");
		for(int i = 0; i < n; i++)
		{
			str[i] = new String(s.next());
		}
		for(int i = 0; i < n; i++)
		{
			for(int j = i+1; j < n; j++)
			{
				if(str[i].compareTo(str[j])>0)
				{
					String temp = str[i];
					str[i] = str[j];
					str[j] = temp;
				}
			}
		}
		System.out.println("Sorted list of strings is:");
		for(int i = 0; i < n ; i++)
```

```
                {
                        System.out.println(str[i]);
                }
        }
}
class nameascending
{
        public static void main(String[] args)
        {
                Sorting obj = new Sorting();
                obj.sortStrings();
        }
}
```

Output:

Enter the value of n:

3

Enter strings:

jaya

bhavani

keerthi

Sorted list of strings is:

bhavani

jaya

Keerthi

VIVA QUESTIONS:

1. Which Java class is used to read input from the user in this program?
2. What method is used to compare two strings in Java?
3. What is sorting in Java?
4. What is the difference between sorting arrays and sorting lists?

PROGRAM-10

10. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

Aim: To Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area ().

Source Code:

```
abstract class Shape
{
abstract void numberOfSides();
}
class Trapezoid extends Shape
{
void numberOfSides()
{
System.out.println(" Trapezoidal has four sides");
}
}
class Triangle extends Shape
{
void numberOfSides()
{
System.out.println("Triangle has three sides");
}
}
class Hexagon extends Shape
{
void numberOfSides()
{
System.out.println("Hexagon has six sides");
}
```

```
}
class ShapeDemo
{
public static void main(String args[ ])
{
Trapezoid t=new Trapezoid();
Triangle r=new Triangle();
Hexagon h=new Hexagon();
Shape s;
s=t;
s.numberOfSides();
s=r;
s.numberOfSides();
s=h;
s.numberOfSides();
}
}
```

Output:

```
D:\JAVA\bhavani>java ShapeDemo

Trapezoidal has four sides

Triangle has three sides

Hexagon has six sides
```

VIVA QUESTIONS:

1. What is the difference between an abstract class and an interface in Java?
2. What is inheritance in Java?
3. What is polymorphism in Java?
4. Explain the concept of abstraction?
5. What is the advantage of using an abstract class instead of an interface in this program?

PROGRAM-11

11. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Aim: To write a java program to that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Source code:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Add1 extends Applet implements ActionListener
{
String msg;
TextField num1, num2, res;
Label l1, l2, l3;
Button div;
public void init()
{
l1 = new Label("Number 1");
l2 = new Label("Number 2");
l3 = new Label("result");
num1 = new TextField(10);
num2 = new TextField(10);
res = new TextField(30);
div = new Button("DIV");
div.addActionListener(this);
add(l1);
```

```
add(num1);
add(l2);
add(num2);
add(l3);
add(res);
add(div);
}
public void actionPerformed(ActionEvent ae)
{
String arg = ae.getActionCommand();
if (arg.equals("DIV"))
{
String s1 = num1.getText();
String s2 = num2.getText();
int num1 = Integer.parseInt(s1);
int num2 = Integer.parseInt(s2);
if (num2 == 0)
{
msg = "Arithemetic Exception ";
repaint();
}
else if ((num1 < 0) || (num2 < 0))
{
msg = "NumberFormat Exception";
repaint();
}
else
{
int num3 = num1 / num2;
msg = String.valueOf(num3);
}
res.setText(msg);
}
}
```

```
public void paint(Graphics g)
{
//g.drawString(msg, 30, 70);
}
}
```

APPLET.HTML

```
<html>
<head>
</head>
<body>
/*<applet code="Add1.class"width=350 height=300>
</applet>*/
</body>
</html>
```

Output:

VIVA QUESTIONS:

1. What is the difference between an applet and a standalone Java application?
2. What is applet?
3. How to compile an applet?
4. How to execute an applet?
5. What is the default layout of an applet?

PROGRAM-12

12. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

Aim: To Write a java program that implements a multi-thread application that has three threads

Source Code:

```
import java.util.Random;
class RandomNumberThread extends Thread {
	public void run() {
		Random random = new Random();
		for (int i = 0; i < 10; i++) {
			int randomInteger = random.nextInt(100);
			System.out.println("Random Integer generated : " + randomInteger);
			if((randomInteger%2) == 0) {
				SquareThread sThread = new SquareThread(randomInteger);
				sThread.start();
			}
			else {
				CubeThread cThread = new CubeThread(randomInteger);
				cThread.start();
			}
			try {
				Thread.sleep(1000);
			}
			catch (InterruptedException ex) {
				System.out.println(ex);
			}
		}
	}
}

class SquareThread extends Thread {
	int number;
```

```
        SquareThread(int randomNumbern) {
                number = randomNumbern;
        }
        public void run() {
                System.out.println("Square of " + number + " = " + (number * number));
        }
}
class CubeThread extends Thread {
        int number;
        CubeThread(int randomNumber) {
                number = randomNumber;
        }
        public void run() {
                System.out.println("Cube of " + number + " = " + number * number * number);
        }
}
public class MultiThreadingTest {
        public static void main(String args[]) {
                RandomNumberThread rnThread = new RandomNumberThread();
                rnThread.start();
        }
}
```

Output:

```
Random Integer generated : 52
Square of 52 = 2704
Random Integer generated : 23
Cube of 23 = 12167
Random Integer generated : 18
Square of 18 = 324
Random Integer generated : 14
Square of 14 = 196
Random Integer generated : 28
Square of 28 = 784
Random Integer generated : 38
```

Square of 38 = 1444

Random Integer generated : 83

Cube of 83 = 571787

Random Integer generated : 54

Square of 54 = 2916

Random Integer generated : 52

Square of 52 = 2704

Random Integer generated : 69

Cube of 69 = 328509

VIVA QUESTIONS:

1. What is the difference between a local variable and an instance variable in Java?
2. What is local variable?
3. What is instance variable?
4. What is the difference between a blocked thread and a suspended thread in Java?
5. What is final keyword?
6. What is the difference between a process and a thread in Java?

PROGRAM-13

13. Write a java program that correct implements of producer consumer program.

Aim: To Write a java program that correct implements of producer consumer program

Source Code:

```
class Q
{
int n;
boolean valueSet=false;
synchronized int get()
{
if(!valueSet)
try
{
wait();
}
catch(InterruptedException e)
{
System.out.println("Interrupted Exception caught");
}
System.out.println("Got:"+n);
valueSet=false;
notify();
return n;
}
synchronized void put(int n)
{
if(valueSet)
try
{
wait();
}
catch(InterruptedException e)
{
System.out.println("Interrupted Exception caught");
```

```
}
this.n=n;
valueSet=true;
System.out.println("Put:"+n);
notify();
}
}
class Producer implements Runnable
{
Q q;
Producer(Q q)
{
this.q=q;
new Thread(this,"Producer").start();
}
public void run()
{
int i=0;
while(true)
{
q.put(i++);
}
}
}
class Consumer implements Runnable
{
Q q;
Consumer(Q q)
{
this.q=q;
new Thread(this,"Consumer").start();
}
public void run()
{
```

```
while(true)
{
q.get();
}
}
}
class ProducerConsumer
{
public static void main(String[] args)
{
Q q=new  Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-c to stop");
}
}
```

Output:

```
Put:1
Got:1
Put:2
Got:2
Put:3
Got:3
Put:4
Got:4
Put:5
Got:5
Put:6
Got:6
Put:7
Got:7
Put:8
Got:8
Put:9
```

Got:9

Put:10

Got:10……..

VIVA QUESTIONS:

1. What is the difference between wait() and sleep() in Java?
2. What is the purpose of the notify() method in the Q class?
3. What is Exception Handling?
4. Explain Multi-Threading?
5. What are advantages of Exception handling in java?

PROGRAM-14

14. Write a program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.

Aim: To Write a program that creates three threads

Source Code:

```
class A extends Thread
{ public void run()
 {
try
{
for(int i=1;i<=10;i++)
{
sleep(1000);
System.out.println("good morning");
}
}
catch(Exception e)
{
System.out.println(e);
 }
 }
 }
 class B extends Thread
{
public void run()
{
try
{
for(int j=1;j<=10;j++)
{
sleep(2000);
System.out.println("hello");
}
```

```
}
catch(Exception e)
{
System.out.println(e);
 }
}
}
class C extends Thread
{
 public void run()
{
try
{
for(int k=1;k<=10;k++)
{
sleep(3000);
System.out.println("welcome");
 }
}
catch(Exception e)
 {
System.out.println(e);
}
}
}
class threaddemo
 {
public static void main(String args[])
 {
 A a1=new A();
 B b1=new B();
C c1=new C();
a1.start();
b1.start();
```

```
c1.start();
}
 }
```

Output:

```
good morning
hello
good morning
welcome
good morning
hello
good morning
good morning
welcome
hello
good morning
good morning
hello
good morning
welcome
good morning
hello
good morning
welcome
hello
hello
welcome
hello
welcome
hello
hello
welcome
welcome
welcome
welcome
```

VIVA QUESTIONS:

1. What is synchronization in Java?
2. How do you catch and handle an exception in a thread?
3. What is the purpose of the start() method in a thread?
4. What is a deadlock in Java?
5. What is the difference between wait() and sleep() in Java?

PROGRAM-15

15.Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

Aim: To write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

Source Code:

```
import java.io.File;
class FileDemo {
static void p(String s) {
System.out.println(s);
}
public static void main(String args[ ]) {
File f1 = new File(args[0]);
p("File Name: " + f1.getName());
p("Path: " + f1.getPath());
p("Abs Path: " + f1.getAbsolutePath());
p("Parent: " + f1.getParent());
p(f1.exists() ? "exists" : "does not exist");
p(f1.canWrite() ? "is writeable" : "is not writeable");
p(f1.canRead() ? "is readable" : "is not readable");
p("is " + (f1.isDirectory() ? "" : "not" + " a directory"));
p(f1.isFile() ? "is normal file" : "might be a named pipe");
p(f1.isAbsolute() ? "is absolute" : "is not absolute");
p("File last modified: " + f1.lastModified());
p("File size: " + f1.length() + " Bytes");
} }
```

Output:

```
D:\JAVA\bhavani>java FileDemo Fib.java
File Name: Fib.java
Path: Fib.java
Abs Path: D:\JAVA\bhavani\Fib.java
Parent: null
```

exists

is writeable

is readable

is not a directory

is normal file

is not absolute

File last modified: 1708752305106

File size: 738 Bytes

VIVA QUESTIONS:

1. What is the purpose of the File class in Java?
2. What is the difference between a file's path and its absolute path?
3. What is the difference between a file system and a file in Java?
4. How to create a new file in Java?
5. How to delete a file in Java?

PROGRAM-16

16.Write a Java Program that displays the number of characters, lines and words in a text file.

Aim: To Write a Java Program that displays the number of characters, lines and words in a text file.

Source Code:

```
import java.io.*;
public class FileStat {
public static void main(String args[ ])throws IOException {
long nl=0,nw=0,nc=0;
String line;
BufferedReader br=new BufferedReader(new FileReader(args[0]));
while ((line=br.readLine())!=null) {
nl++;
nc=nc+line.length();
int i=0;
boolean pspace=true;
while (i<line.length()) {
char c=line.charAt(i++);
boolean cspace=Character.isWhitespace(c);
if (pspace&&!cspace)
nw++;
pspace=cspace;
}
}
System.out.println("Number of Characters"+nc);
System.out.println("Number of Characters"+nw);
System.out.println("Number of Characters"+nl);
}}
```

Output:

```
D:\JAVA\bhavani>javac FileStat.java
D:\JAVA\bhavani>java FileStat abc.txt
Number of Characters553
Number of Characters54
Number of Characters23
```

VIVA QUESTIONS:

1. What is the difference between a char and a String in Java?
2. What is the purpose of the try-catch block in Java?
3. What is Buffer reader?
4. What is Buffer Writer?
5. Why the main method is static in java?

PROGRAM-17

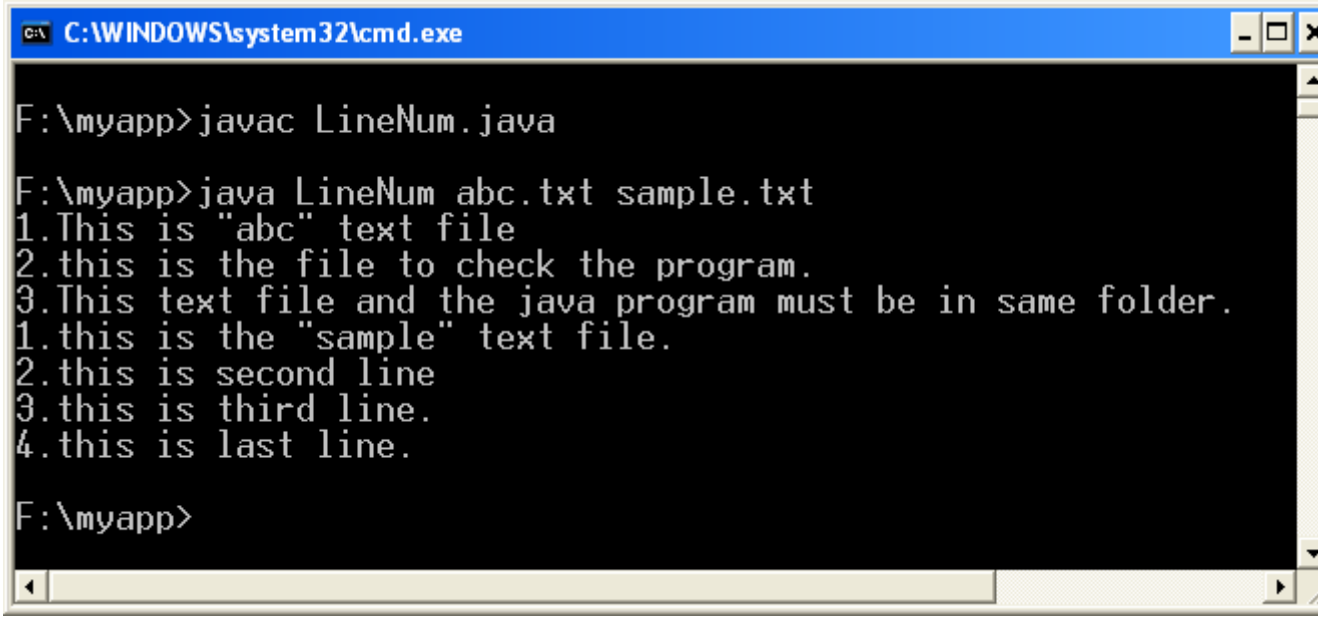
17.Write a java program that reads a file and displays the file on the screen with line number before each line.

Aim: To Write a Java Program that reads a file and displays the file on the screen with line number before each line.

Source Code:

```
import java.io.*;
class LineNum{
public static void main(String args[]){
String thisline;
for(int i=0;i<args.length;i++)
{
try{
LineNumberReader br=new LineNumberReader(new FileReader(args[i]));
while((thisline=br.readLine())!=null)
{
System.out.println(br.getLineNumber()+"."+thisline);
}
}catch(IOException e){
System.out.println("error:"+e);
} } } }
```

Input & Output :-



VIVA QUESTIONS:

1.What is buffer reader class?
2. What is scanner class.
3.Why buffer reader is used to read line by line?

Experiment(s) Beyond Syllabus

1.Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

Program:-

```
public class Prog1
{
public static void main(String[] args)
{
System.out.println("\n Prog. is showing even no");
for(int i=2;i<=20;i++)
{
if(i%2==0)
{
System.out.print("\n "+i);
}
}
}
}
```

Compile:-

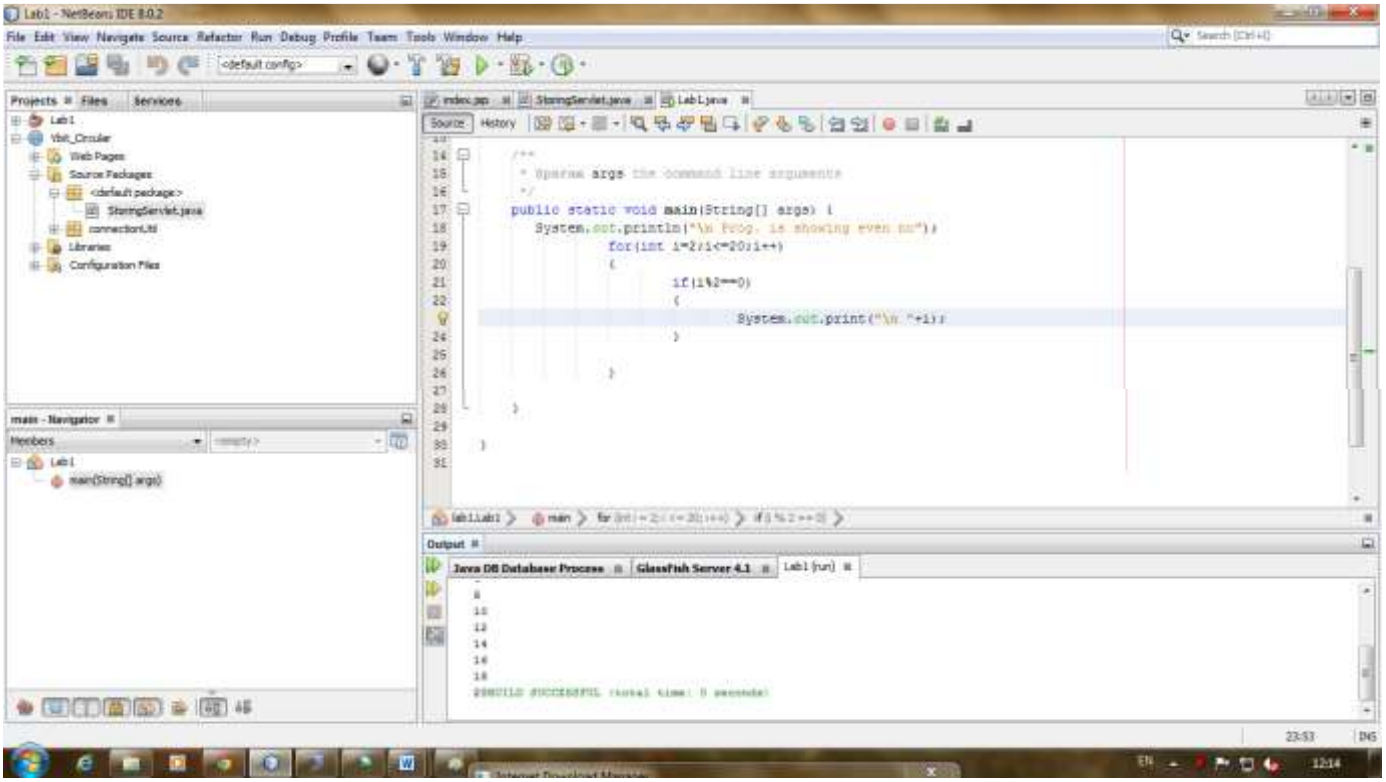
D:>javac Prog1.java

Run:-

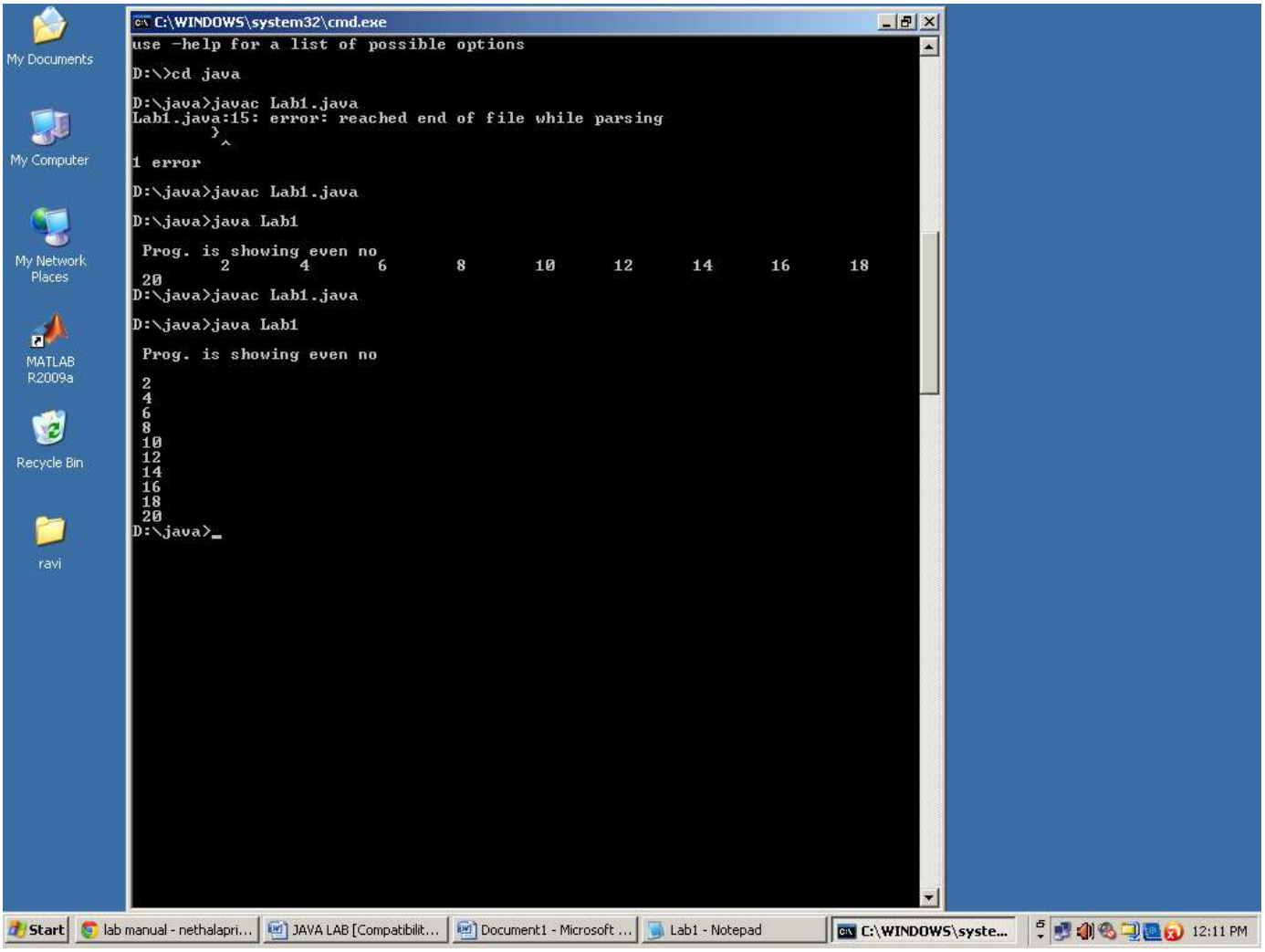
D:>java Prog1

Output:-

In Netbeans IDE:-



In command Prompt



VIVA QUESTIONS:

1.How to install Netbeans?

2. What is the difference between Netbeans and command path?