



**Progressive Education Society's  
Modern College Of Arts, Science & Commerce (Autonomous) Ganeshkhind,  
Pune – 411016**

**S.Y.B.C.A. (Science) Semester-IV**

**(CBCS Pattern)**

**23-BCA-244**

**Core Java Laboratory**

**Lab Book**

**Academic Year 2023–2024**

Name:

College Name:

Roll No.: Seat No:

Academic Year:

## *CERTIFICATE*

This is to certify that Mr./Ms. \_\_\_\_\_  
has successfully completed **SYBCA (Science) 23-BCA-244**  
**Core Java Laboratory Semester IV** course in year \_\_\_\_\_ and  
his/her seat no. is \_\_\_\_\_. He / She has scored \_\_\_\_\_  
marks out of 15.

Instructor

HOD

Internal Examiner

External Examiner

## INDEX

### Assignment Completion Sheet

Core Java Laboratory			
Assignment No	Assignment Name	Marks (out of 5)	Teacher Sign
1	Java Tools and IDE, Simple java programs.		
2	Classes and Objects		
3	Array of Objects and Packages		
4	Inheritance and Interfaces		
5	Exception And File Handling		
6	GUI Designing, Event Handling		
Total (Out of 30)			
Total (Out of 15)			

Assignment No. 1	
Assignment Name :	Java Tools and IDE, Simple java programs.

### Prerequisites:-

- OOP's Concepts
- Keywords used in Java
- Structure of Java Program

Java is an extensively used programming language. It has been used by many programmers from beginners to experts. If we look into the phases of program development, we come across many phases such as:

### Step 1: Creation of a Java program:-

Creating a Java program means writing a Java program on any editor or IDE. After creating a Java program, provide a .java extension to file. It signifies that the particular file is a Java source code. Also, whatever progress we make from writing, editing, storing and providing .java extension to a file, it basically comes under the creation of a Java program.

### Step 2: Compiling a Java Program

Our next step after the creation of the program is the compilation of the Java program. Generally Java programs which we have created in step 1 with a .java extension, have been compiled by the compiler. Suppose if we take an example of a program, say Welcome.java, when we want to compile this we use a command such as javac. After opening command prompt or shell console we compile Java

program with .java extension as:

```
javac Welcome.java
```

After executing the javac command, if no errors occur in our program we get a .class file which contains the bytecode. It means that the compiler has successfully converted Java source code to bytecode. Generally bytecode is used by the JVM to run a particular application. The final bytecode created is the executable file which only JVM understands. As Java compiler is invoked by javac command, the JVM is invoked by java command. On the console window you have to type:

```
java Welcome
```

This command will invoke the JVM to take further steps for execution of Java program.

### Step 3: Program loading into memory by JVM:-

JVM requires memory to load the .class file before its execution. The process of placing a program in memory in order to run is called as Loading. There is a class loader present in the JVM whose main functionality is to load the .class file into the primary memory for the execution. All the .class files required by our program for the execution is been loaded by the class loader into the memory just before the execution.

### Step 4: Bytecode Verification by JVM:-

In order to maintain security of the program JVM has bytecode verifier. After the classes are loaded in to memory, bytecode verifier comes into picture and verifies bytecode of the loaded class in order to maintain security. It check

whether bytecodes are valid. Thus it prevent our computer from malicious viruses and worms.

### Step 5: Execution of Java program: -

Whatever actions we have written in our Java program, JVM executes them by interpreting bytecode. JVM uses JIT compilation unit to which we even call just-in-time compilation.

### JVM (Java Virtual Machine):-

JVM, i.e., Java Virtual Machine. JVM is the engine that drives the Java code. Mostly in other Programming Languages, compiler produce code for a particular system but Java compiler produce Bytecode for a Java Virtual Machine. When we compile a Java program, then bytecode is generated. Bytecode is the source code that can be used to run on any platform. Bytecode is an intermediary language between Java source and the host system. It is the medium which compiles Java code to bytecode which gets interpreted on a different machine and hence it makes it Platform/Operating system independent.

### A Simple Java Program

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

**class** keyword is used to declare a class in Java.

- **public** keyword is an access modifier that represents visibility. It means it is visible to all.

- **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main () method is executed by the JVM, so it doesn't require creating an object to invoke the main () method. So, it saves memory.

- **void** is the return type of the method. It means it doesn't return any value.

- **main** represents the starting point of the program.

- **String[] args** or **String args[]** is used for command line argument.

**System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class.

To compile the program the command is

```
javac Test.java
```

To execute the program the command is

```
java Test
```

### Different ways of reading input from console in Java -

#### 1. Using command line argument:

The java command-line argument is an argument i.e. passed at the time of running the java program. The arguments

passed from the console can be received in the java program and it can be used as an input.

```
public class Test
{
    public static void main(String[] args)
    {
        int a,b,c;
        a = Integer.parseInt(args[0]);
        b = Integer.parseInt(args[1]);
        c = a+b;
        System.out.println("Addition:" +c);
    }
}
javac Test.java
java Test 10 20
```

## 2. Using Buffered Reader Class

This is the Java classical method to take input. This method is used by wrapping the System.in (standard input stream) in an InputStreamReader which is wrapped in a BufferedReader, we can read input from the user in the command line.

```
import java.io.*;
public class ConsoleInput
{
    public static void main (String[] args) throws IOException
    {
        int rollNumber;
        String name;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the roll number: ");
        rollNumber = Integer.parseInt(br.readLine());
        System.out.println(" Enter the name: ");
        name = br.readLine();
        System.out.println(" Roll Number = " + rollNumber);
        System.out.println(" Name = " + name);
    }
}
```

## 3. Using Scanner Class

This is probably the most preferred method to take input. The main purpose of the Scanner class is to parse primitive types and strings using regular expressions, however it also can be used to read input from the user in the command line.

```
import java.util.Scanner;
public class ScannerTest
{
    public static void main(String args[])throws Exception
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your rollno and name :");
```

```
int rollno=sc.nextInt();  
String name=sc.next();  
System.out.println("Rollno:"+rollno+" Name:"+name);  
sc.close();  
}  
}
```

## Lab Assignment

### SET A

1. Write a program to calculate perimeter and area of rectangle. (hint :  $\text{area} = \text{length} * \text{breadth}$  ,  $\text{perimeter} = 2 * (\text{length} + \text{breadth})$ ) (Scanner class) .

2. accept a number as commandline argument and check if it is divisible by 5 (Commandline argument)

3. Write a program to accept n numbers and find their sum and average. (scanner class)

Set B

1. Write a program to accept a number from user. Check whether number is prime or not. Use BufferedReader class for accepting input from user.



2. Accept a number and check if it is odd or even.(Bufferreader class)

3. program to print the following pattern .

```
1
1 2
1 2 3
```

## SET C

1. Write a Java program to print the sum of elements of the array. Also display array elements in ascending order.
2. Write a Java Program to reverse a number . Accept number using Command Line .

**Signature of the instructor:** \_\_\_\_\_ **Date:** \_\_\_\_\_

### Assignment Evaluation

0: Not Done [ ]

1: Incomplete [ ]

2: Late Complete [ ]

3: Needs Improvement [ ]

4: Complete [ ]

5: Well done [ ]