JAVA PROGRAMMING

LAB MANUAL

(2022-23)

DEPARTMENT OF AI&ML AND IOT

SMT.INDRA GANDHI COLLEGE OF ENGINEERING

Sector 8, Plot 1, Ghansoli, Navi Mumbai, Maharashtra 400701

Prepared By
Prof.Bhagyalakshmi

CSL304 SKILL BASED LAB COURSE:-
OBJECT ORIENTED PROGRAMMING WITH JAVA LABORATORY

Lab Objectives:

1. To learn the basic concepts of object-oriented programming.
2. To study JAVA programming language
3. To study various concepts of JAVA programming like multithreading, exception Handling Packages, etc.
4. To explain components of GUI based programming.

Lab Outcomes:

At the end of the course, the students should be able to:

1. To apply fundamental programming constructs.
2. To illustrate the concepts of packages, classes and objects.
3. To elaborate the concept of strings arrays and vectors.
4. To implement the concept of inheritance and interfaces.
5. To implement the concept of exception handling and multithreading.
6. To develop GUI based application.

Lab Code	Lab Name	Credits
CSL304	Skill based Lab Course: Object Oriented Programming with Java	2

Prerequisite: Structured Programming Approach	
Lab Objectives:	
1	To learn the basic concepts of object-oriented programming
2	To study JAVA programming language
3	To study various concepts of JAVA programming like multithreading, exception Handling, packages, etc.
4	To explain components of GUI based programming.
Lab Outcomes: At the end of the course, the students should be able to	
1	To apply fundamental programming constructs.
2	To illustrate the concept of packages, classes and objects.
3	To elaborate the concept of strings arrays and vectors.
4	To implement the concept of inheritance and interfaces.
5	To implement the concept of exception handling and multithreading.
6	To develop GUI based application.

Module		Detailed Content	Hours
1		**Introduction to Object Oriented Programming**	**2**
	1.1	OOP concepts: Objects, class, Encapsulation, Abstraction, Inheritance, Polymorphism, message passing.	
	1.2	Java Virtual Machine	
	1.3	Basic programming constructs: variables, data types, operators, Unsigned right shift operator, expressions, branching and looping.	
2		**Class, Object, Packages and Input/output**	**6**
	2.1	Class, object, data members, member functions Constructors, types, static members and functions Method overloading Packages in java, types, user defined packages Input and output functions in Java, Buffered reader class, scanner class	
3		**Array, String and Vector**	**3**
	3.1	Array, Strings, String Buffer, Vectors	
4		**Inheritance**	**4**
	4.1	Types of inheritance, Method overriding, super, abstract class and abstract method, final, Multiple inheritance using interface, extends keyword	
5		**Exception handling and Multithreading**	**5**
	5.1	Exception handling using try, catch, finally, throw and throws, Multiple try and catch blocks, user defined exception Thread lifecycle, thread class methods, creating threads using extends and implements keyword.	
6		**GUI programming in JAVA**	**6**
	6.1	Applet and applet life cycle, creating applets, graphics class functions, parameter passing to applet, Font and color class. Event handling using event class AWT: working with windows, using AWT controls for GUI design Swing class in JAVA	

		Introduction to JDBC, JDBC-ODBC connectivity, JDBC architecture.	

Textbooks:	
1	Herbert Schildt, „JAVA: The Complete Reference“, Ninth Edition, Oracle Press.
2	E. Balagurusamy, „Programming with Java“, McGraw Hill Education.

References:	
1	Ivor Horton, “Beginning JAVA“, Wiley India.
2	Dietal and Dietal, “Java: How to Program“, 8th Edition, PHI .
3	“JAVA Programming“, Black Book, Dreamtech Press.
4	“Learn to Master Java programming“, Staredu solutions

Digital material:	
1	www.nptelvideos.in
2	www.w3schools.com
3	www.tutorialspoint.com
4	https://starcertification.org/Certifications/Certificate/securejava

Suggested List of Programming Assignments/laboratory Work:	
Sr. No.	**Name of the Experiment**
1	Programs on Basic programming constructs like branching and looping
2	Program on accepting input through keyboard.
3	Programs on class and objects
4	Program on method and constructor overloading.
5	Program on Packages
6	Program on 2D array, strings functions
7	Program on String Buffer and Vectors
8	Program on types of inheritance
9	Program on Multiple Inheritance
10	Program on abstract class and abstract methods.
11	Program using super and final keyword
12	Program on Exception handling
13	Program on user defined exception
14	Program on Multithreading
15	Program on Graphics class
16	Program on applet class
17	Program to create GUI application
18	Mini Project based on the content of the syllabus (Group of 2-3 students)

Term Work:	
1	Term work should consist of 15 experiments.
2	Journal must include at least 2 assignments
3	Mini Project based on the content of the syllabus (Group of 2-3 students)
4	The final certification and acceptance of term work ensures that satisfactory performance of Laboratory work and minimum passing marks in term work.
5	Total 50-Marks (Experiments: 15-marks, Attendance: 05-marks, Assignments: 05-marks, Mini Project: 20-marks, MCQ as a part of lab assignments: 5-marks)

Oral & Practical exam
Based on the entire syllabus of CSL 304: **Skill based Lab Course: Object Oriented Programming with Java**

SL304 SKILL BASED LAB COURSE:-

OBJECT ORIENTED PROGRAMMING WITH JAVA LABORATORY

INDEX

S.No	List of programs	Pg.No.
	Introduction to OOPM	1
	Sample Programs	2
1	Programs on Basic programming constructs like branching and looping	3
2	Program on accepting input through keyboard.	4
3	Programs on class and objects	
4	Program on method and constructor overloading.	
5	Program on Packages	
6	Program on 2D array, strings functions	
7	Program on String Buffer and Vectors	
8	Program on types of inheritance	
9	Program on Multiple Inheritance	
10	Program on abstract class and abstract methods.	
11	Program using super and final keyword	
12	Program on Exception handling	
13	Program on user defined exception	
14	Program on Multithreading	
`15	Program on Graphics class	
16	Program on applet class	
17	Program to create GUI application	
18	Mini Project based on the content of the syllabus (Group of 2-3 students)	

INTRODUCTION TO OOP

Object-oriented programming (OOP) is a computer science term used to characterize a programming language that began development in the 1960's. The term 'object-oriented programming' was originally coined by Xerox PARC to designate a computer application that describes the methodology of using objects as the foundation for computation. By the 1980's, OOP rose to prominence as the programming language of choice, exemplified by the success of C++. Currently, OOPs such as Java, J2EE, C++,C=, Visual Basic.NET, Python and java Script are popular OOP programming languages that any career-oriented Software Engineer or developer should be familiar with.

OOP is widely accepted as being far more flexible than other computer programming languages. OOPs use three basic concepts as the fundamentals for the Abstraction, Polymorphism, Event Handling and Encapsulation are also significant concepts within object-oriented programming languages that are explained in online tutorial describing the functionality of each concept in detail.

The java platform is undoubtedly fast moving and comprehensive. Its many application programming interfaces (APIs) provide a wealth of functionality for all aspects of application and system-level programming. Real-world developers never use one or two APIs to solve a problem, but bring together key functionality spanning a number of APIs, Knowing which APIs you need,

which parts of which APIs you need, and how the APIs work together to create the best solution can be a daunting task.

Simple Programs

Java Basic.

1.Write a Program to print the text "Welcome to World of Java". Save it with name Welcome.java in your folder.

```
Class Welcome { public static void main (String args[])
 {
 System.out.println ("welcome to world of Java");
 }
 }
```

2.Write a Program to print the area of triangle. Save it with name Area.java in your folder.

```
class Area { public static void main(String args[])
{
 int height =10, base=6;
float area=0.5F*base* height;
 System.out.println("area of triangle = "+area);
 }
 }
```

3. Write a java Program to check the number is Prime or not.

```
 Import java.util.Scanner;
class Prime { public static void main(String arr[])
{
 int c; Scanner in=new Scanner(System.in);
 System.out.println("Enter the number to be tested for prime ");
int n=in.nextInt();
 for ( c = 2 ; c <= n - 1 ; c++ )
{
 if ( n%c == 0 )
 {
System.out.println(n+">>>>> not prime");
 break;
 }
 }
 if ( c == n )
System.out.println(n+ ">>>>Number is prime.");
 }
  }
```

Program 1
AIM:- To understand programming constructs like branching and looping

PROBLEM STATEMENT:-

Write a program to print Even number

THEORY:-

Branching statements give us code which is optionally executable, depending on the outcome of certain tests or you can say certain cases which we can define.
Looping statements are used to repetition of a section of code a number of times or until a
Condition has been fulfilled.

```
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:-

2,4,6,8,01,12,14,16,18,20

CONCLUSION:-

We understood programming constructs like branching and looping

Write a program based on pattern generation

```
Package javaPractical;
import java.util.Scanner;

public class PatternGeneration{
public ststic void main(String[] args){

Scanner sc= new Scanner(System.in);
System.out.println(“Eneter no. of rows”);

int n =sc nextInt();
for(int i-0;i<n;i++){
for(int j=n-1;j>i,j--){
System.out.println(“ ”);
}

for(int j=0;j<=I; j++){
System.out.print(“*“);
}

for(int j=0;j<I;j++){
System.out.print(“*”)
}
System.out.println();
}
}
}
```

OUTPUT:-

```
*
**
***
****
```

CONCLUSION:-

We understood programming constructs like branching and looping

Program 2

AIM:- To understand accepting input through keyboard in java program

PROBLEM STATEMENT:- Write a simple input program

Write a program to create a class Student with data ‘name, city and age’ along with method printData to display the data. Create the two objects s1 ,s2 to declare and access the values.

```
class Student
{
 String name, city; int age; staticint m; voidprintData()
{
 System.out.println("Student name = "+name);
System.out.println("Student city = "+city);
System.out.println("Student age = "+age);
}
 }
 classStest
{
 public static void main(String args[])
 {
11 Student s1=new Student();
 Student s2=new Student();
 s1.name="Amit";
s1.city="Dehradun";
 s1.age=22;
s2.name="Kapil";
s2.city="Delhi";
 s2.age=23;
 s2.printData();
 s1.printData();
 s1.m=20;
s2.m=22;
Student.m=27;
System.out.println("s1.m = "+s1.m);
 System.out.println("s2.m = "+s2.m);
 System.out.println("Student.m ="+Student.m);
 }
 }
```

Write a program to create a class Student2 along with two method getData(),printData() to get the value through argument and display the data in printData. Create the two objects s1 ,s2 to declare and access the values from class STtest.

```
 class Student2
{ private String name, city;
privateint age;
 public void getData(String x, Stringy, int t)
 {
 name=x; city=y; age=t;
 }
 public void printData()
 {
System.out.println("Student name ="+name);
System.out.println("Student city ="+city);
 System.out.println("Student age ="+age);
 }
}
classSTtest { public static void main(String args[])
{
Student2 s1=new Student2();
 Student2 s2=new Student2();
 s2.getData("Kapil","Delhi",23);
s2.printData();
 s1.getData("Amit","Dehradun",22);
s1.printData();
 }
 }
```

Program No: 03
AIM:- To understand class and object in java
PROBLEM STATEMENT:-
Write a program based on class and object
THEORY:-

 Classes and Objects are basic concepts of Object Oriented Programming that revolve around real life entities.
 A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.
 In Java, the new keyword is used to create new objects.
 Declaration − A variable declaration with a variable name with an object type.
Instantiation − The 'new' keyword is used to create the object.
Initialization − The 'new' keyword is followed by a call to a constructor.

Program code:

```
Package javaPractical3;
Import java.util.Scanner;
Class Employee{
Scanner sc =new Scanner(System.in);
Int empId;
String empContactNumber,empName;
Void enterInfo(){
System.out.println(“Enter Employee Information\n”);
System.out.println(“Enter employee ID”);
Empid=sc.nextInt();
Sc.nextLine();
System.out.println(“Enter employee Name”);
empName=sc.nextLine();
System.out.println(“Enter Contact Name”);
empContactNUmber=sc.nextLine();
System.out.println(“\n\n”);
}
Void showInfo(){
System.out.println(“Employee ID: “+empID);
System.out.println(“Employee Name: “+empName);
System.out.println(“Emp Contact no: “+empContactNumber+”\n”);
}
}
Public class EmployeeInfor{
Public static void main(String[] args){
Employee emp1=new Employee();
emp1.enterInfo();
Employee emp2=new Employee();
emp1.enterInfo();
emp1.showInfo();
emp2.showInfo();
}
}
```

OUTPUT:-

```
Enter Employee Information
Enter employee ID
101
Enter employee Name xyz
```

Enter Contact No.
1234567890
Enter Employee Information
Enter employee ID
102
Enter employee Name
abc
Enter Contact No.
9876543210
Employee ID :101
Employee Name :xyz
Emp Contact no : 1234567890

Employee ID :102
Employee Name :abc
Emp Contact no : 9876543210
CONCLUSION:-
We understand class and object in java

Program No: 04
AIM:- To understand method and constructor overloading.
PROBLEM STATEMENT:-
Write a program based on method and constructor overloading.
THEORY:-
 Constructors must have the same name as the class within which it is defined while it is not necessary for the method in Java.
 Constructors do not return any type while method(s) have the return type or void if does not return any value.
Program code:

```
Package javaPractical4;
Class AreaClass{
int side1;
int side2;
int side3;
int side4;
public AreaClass(int a, int c){
side1=a;
side2=b;
side3=c;
System.out.println(“It is Triangle with sides as \n” +a+ units, “+b” units, +c+ units”;
}
Public AreaClass(int a){
Side1=a;
System.out.println(“It is Square with side as \n” +a+ units.”);
}
Public AreaClass(int a, int b){
Side1=a;
Side2=b;
System.out.println (“It is Rectangle with side as \n” +a+ units,”+b+” units.”);
}
void getArea(int a ){
double are=a*a;
System.out.println(“Area of Square is “ +area+” sq.units”);
}
```

```
void getArea(double base, double side) {
double area=0.5*base*(side*1.73)/2;
System.out.println(“Area of Triangle is “ +area+” sq.units”);
}
void getArea(int a, int b ){
double area=a*b;
System.out.println(“Area of Rectangle is “ +area+” sq.units”);
}
}
Public class Area{
Public static void main(String[] args){
AreaClaa shape1=new AreaClass(5);
Shape1.getrArea(5);
System.out.println();
AreaClass shape2= new AreaClass(5,6);
Shape2.getArea(5,6);
System.out.println();
AreaClass shape3=new AreaClass(20, 20,20);
Shape3.getArea(20,20);
System.out.println ()
}
}
```

OUTPUT:-
It is a Square with each side as 5 units.
Area of square is 25.0sq.units

It is a Rectangle with each side as 5 units.6 units.
Area of square is 30.0sq.units

It is a Triangle with each side as 20 units, 20 units
Area of square is 400.0sq.units

CONCLUSION:
We understood method and constructor overloading.

Program No: 05
AIM: - To understand packages in java.
PROBLEM STATEMENT:-
Write a program based on packages.
THEORY:-
 A package in Java is used to group related classes. Think of it as a folder in a file directory. We use packages to avoid name conflicts, and to write a better maintainable code.
 Using one import statement, we may import only one package or a class.
 Using an import statement, we cannot import a class directly, but it must be a part of a package.
 A program may contain any number of import statements.

Program code:

```
Package inheritance.BasicCalculator;
Class Calci{
Int z;
Public void multiplication(int x, int y)
{
z=x*y;
```

```
System.out.println("The product of the give=n numbers:"+z);
}
Public void addition (int x, int y)
{
z=x-y;
System.out.println("The product of the given numbers:"+z);
}
}
Public class Calculation extends Calci
{
Public static void main(String args[]){
int a =20, b=10;
Calculation demo=new Calculation();
demo.multiplication(a,b);
}
}
```

OUTPUT:-
The product of the given number: 5
The product of the given number: 6
The product of the given number: 3

CONCLUSION:-
We understood packages in java
Experiment No: 06
AIM: - To understand concept of 2D array and String Function.
PROBLEM STATEMENT:-
Write a program based on
A) Matrix addition
B) Matrix multiplication
THEORY:-
 Two-dimensional (2D) arrays are indexed by two subscripts, one for the row and one for the column. Each element in the 2D array must by the same type, either a primitive type or object type.
 The string is a collection of characters; an array of a string is an array of arrays of characters. Each string is terminated with a null character. An array of a string is one of the most common applications of two-dimensional arrays.
Program code:

```
Package javaPractical6;
Import java.util.Scanner;
Public class MatrixAddition {
Static void printArray (int arr[][]){
Int r=arr.length;
Int c =arr[0].length;
for(int i=0;i&lt;r;i++){
for(int j=0;j&lt;c;j++){
System.out.println(arr[i[[j]){
}
System.out.porintln()'
}
}
Public static void main(String[] args){
Scanner sc=new Scanner(System.in);
int row,col;
```

```
System.out.println(“enter no of rows and columns for matric=x”);
row=sc.nextInt();
col=sc.nextInt();
int a[ ][ ]=new int [row][col];
int b[ ][ ]=new int [row][col];
int b[ ][ ]=new int [row][col];
System.ouyt println(“enter elements of matrix a”);
for(int i=0;i&lt;row;i++)
for(int j=0;j&lt;col;j++)
a[i][j]=sc.nextInt();
}
}
System.out println(“enter elements of matrix b”);
for(int i=0;i&lt;row;i++)
for(int j=0;j&lt;col;j++)
b[i][j]=sc.nextInt();
}
}
for(int i=0;i&lt;row;i++){
for(int j=0;j&lt;col;j++){
c[i][j]=a[i][j]+b[i][j];
}
}
System.out.println(“Matrix C”);
printArrays(c);
}
}
```

OUTPUT:-

CONCLUSION:-
We understood concept of 2D array and String Function

Program No: 07
AIM: - To understand String buffer and vector.
PROBLEM STATEMENT:-
Write a program based on String Buffer and Vector.
THEORY:-
 Java StringBuffer class is used to create mutable (modifiable) String objects. The StringBuffer class in Java is the same as String class except it is mutable i.e. it can be changed.
 The Vector class is used in Java to store data using the List interface. For instance, a Vector may be used to store a list of products sold at a department store or a list of supplements available at a local drug store.
Program code:

```
Package javaPractical7;
Import java.util.*;
Publiv=c class StringBufferAndVector {
Public static void main(String[] args){
vector &lt;Integer&gt; vector=new Vector&lt;&gt;();
vector.add(12);
vector.add(13);
vector.add(14);
System.out.prinltln(vector);
Scanner sc=new Scanner(System.in);
String str=sc.nextLine();
int len=str.length();
```

```
StringBuffer sb=new StringBuffer();
for(int i=0;i&lt;len;i++){
String c=Character.toString(str.charAt(i));
if(str.charAt(i)==’a’||str.charAt(i)==’e’\\str.charArt==’i;||
str.charAt(i)==’o’||str.charAt==’u’||str.charAt(i)==’y’||
str.charAt(i)==’A’||str.charAt==’E’||str.charAt(i)==’I’||
str.charAt(i)==’O’||str.charAt==’U’||str.charAt(i)==’Y’||{
continuer;
}
sb.append(‘,’);
sb.append(c.toLowerCase());
}
System.out.println(sb.toString());
}
}
```