

BRCM CET, BAHAL



LAB MANUAL

Programming in Java Lab (LC-CSE-327G)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Check list for Lab Manual

S. No.	Particulars
1	Mission and Vision
2	Course Outcomes
3	Guidelines for the student
4	Software & Hardware requirements
5	List of Programs as per University
6	Sample copy of File

Department of Computer Science & Engineering

Vision and Mission of the Department

Vision

To be a Model in Quality Education for producing highly talented and globally recognizable students with sound ethics, latest knowledge, and innovative ideas in Computer Science & Engineering.

MISSION

To be a Model in Quality Education by

M1: Imparting good sound theoretical basis and wide-ranging practical experience to the Students for fulfilling the upcoming needs of the Society in the various fields of Computer Science & Engineering.

M2: Offering the Students an overall background suitable for making a Successful career in Industry/Research/Higher Education in India and abroad.

M3: Providing opportunity to the Students for Learning beyond Curriculum and improving Communication Skills.

M4: Engaging Students in Learning, Understanding and Applying Novel Ideas.

Course: Programming in Java Lab

Course Code: LC-CSE-327G

CO (Course Outcomes)		RBT*- Revised Bloom's Taxonomy
CO1	To Identify of the structure and model of the Java programming language.	L2 (Understand)
CO2	To Identify classes, objects, members of a class and relationships among them needed for a specific problem.	L2 (Understand)
CO3	To Demonstrate the concepts of polymorphism and inheritance.	L3 (Apply)
CO4	To Design Java programs to implement error handling techniques using exception handling.	L6 (Create)
CO5	To Design Java application programs using OOP principles and proper program structuring.	L6 (Create)

CO PO-PSO Articulation Matrices

Course Outcomes (COs)	(POs)												PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2										1	2	3
CO2	2	3	2		2							1	2	2
CO3	2	2	3		2							1	3	2
CO4	2	2	3		2							1	3	2
CO5	2	2	2	3	2							1	3	2

Guidelines for the Students:

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.
4. They should implement the given Program individually.
5. While conducting the experiments students should see that their programs would meet the following criteria:
 - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
 - Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
 - Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs
 - Statements within the program should be properly indented
 - Use meaningful names for variables and functions.
 - Make use of Constants and type definitions wherever needed.
6. Once the experiment(s) get executed, they should show the program and results to the instructors and copy the same in their observation book.
7. Questions for lab tests and exam need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

LIST OF PROGRAMS

1. Write a program to print hello world using java.
2. Write a program for control statement in java.
3. Write a program for function overloading in java.
4. Write a program to show the concept of classes and methods in java.
5. Write a program to implement array in java.
6. Write a program to create applet program.
7. Write a program for multilevel inheritance in java.
8. Program for interface implementation in java.
9. Write a program to show concept of exceptional handling in java.
10. Write a program to implement the concept of swings in java.
11. Write a program that handle all mouse events and show the events name at center of window when a mouse event is fixed in java.
12. Write a program that connects to a database using JDBC and does add, delete ,modify and retrieve operation in java.

PROGRAM 1:

#Write a program to print helloworld in java.

Program :

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

output:

Hello, World!

PROGRAM 2:

#Write a program for control statement in java.

Program :

```
public class Student {  
    public static void main(String[] args) {  
  
        int x = 10;  
        int y = 12;  
  
        if(x+y < 10) {  
            System.out.println("x + y is less than 10");  
        } else {  
            System.out.println("x + y is greater than 20");  
        }  
    }  
}
```

output:

```
x + y is greater than 20
```

PROGRAM 3:

#Write a program to implement the concept of function overloading in java.

Program :

```
class Adder{

    static int add(int a,int b){return a+b;}
    static int add(int a,int b,int c){return a+b+c;}

}

class TestOverloading1 {

    public static void main(String[] args){

        System.out.println(Adder.add(11,11));
        System.out.println(Adder.add(11,11,11));

    }}


```

output:

22

33

PROGRAM 4:

#Write a program to show concept of classes and method in java.

Program :

```
public class Main {

    public void fullThrottle() {
        System.out.println("The car is going as fast as it can!");
    }

    public void speed(int maxSpeed) {
        System.out.println("Max speed is: " + maxSpeed);
    }

    public static void main(String[] args) {
        Main myCar = new Main();
        myCar.fullThrottle();
        myCar.speed(200);
    }
}
```

output:

```
The car is going as fast as it can!
Max speed is: 200
```

PROGRAM 5:

#Write a program to implement array using java.

#Program

```
class Testarray{

    public static void main(String args[]){
    int a[]=new int[5];

    a[0]=10;
    a[1]=20;
    a[2]=70;
    a[3]=40;
    a[4]=50;

    for(int i=0;i<a.length;i++)
    System.out.println(a[i]);
    }}
```

output:

```
10
20
70
40
50
```

PROGRAM6:

#Write a program to create an applet program.

Program :

```
import java.applet.Applet;
import java.awt.Graphics;

public class First extends Applet{

    public void paint(Graphics g){
        g.drawString("welcome to applet",150,150);
    }
}
```

output:

welcome to applet

PROGRAM 7:

#Write a program for multilevel inheritance in java.

Program :

```
class Animal{
    void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
    void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
    public static void main(String args[]){
        BabyDog d=new BabyDog();
        d.weep();
        d.bark();
        d.eat();
    }
}
```

output:

```
weeping...
barking...
eating...
```

PROGRAM 8 :

#Write a program to implement interface in java.

Program :

```
interface printable{  
  
    void print();  
  
}  
  
class A6 implements printable{  
  
    public void print(){System.out.println("Hello");}  
  
  
  
    public static void main(String args[]){  
  
        A6 obj = new A6();  
  
        obj.print();  
  
    }  
  
}
```

output:

Hello

PROGRAM 9:

#Write a program to show the concept of exceptional handling in java.

Program :

```
public class JavaExceptionExample{  
    public static void main(String args[]){  
        try{  
            int data=100/0;  
        }catch(ArithmeticException e){System.out.println(e);}  
        System.out.println("rest of the code...");  
    }  
}
```

output:

```
Exception in thread main java.lang.ArithmeticException:/ by zero  
rest of the code...
```

PROGRAM 10:

#Write a program to implement the concept of swings in java.

Program :

```
import javax.swing.*;

public class FirstSwingExample {

    public static void main(String[] args) {

        JFrame f=new JFrame();

        JButton b=new JButton("click");

        b.setBounds(130,100,100, 40);

        f.add(b);

        f.setSize(400,500);

        f.setLayout(null);

        f.setVisible(true);

    }

}
```

output:

click

