Lab Program - 6

```
public class Student
{
    public int USN;
    public string name;
    public int sem;

public Student ( int USN, String name, int sem)
  {
    this.USN=USN;
     this.name = name;
      this.sem = sem;
}
}
```

```
package cie;

public class Internals extends Student {
        public int[] cie Marks = new Int[5];
public Internals ( int USN, string name, int sem,
                int[] cie Marks)
{
  Super (USN, name, sem);
        this. cie Marks = cie Marks;
```

```
}
}

package see;
import cie;

public class externals extends Students
{
  int[] seeMarks = new int[5]

  public externals ( int USN, String name, int sem,
                     int[]seeMarks)
  {
    super( usn, name, sem);
      this.seeMarks = seeMarks;
  }
}

import see*;
import cie*;

class Main {
  public static void main (String[] args)
  {
```

Date
Page

```
int USN = 123; String name = "Sham"; int sem = 3;
int cie = { 49, 50, 46, 47, 48};
int see = { 91, 100, 89, 91, 90};

Internals in = new internals (USN, name, Sem, cie);
externals ec = new externals (USN, name, Sem, see);

System.out.println ("Name:" + in.name);
System.out.println ("USN:" + ec.USN);
System.out.println ("Sem:" + in.Sem);

int total = 0;
for (int i=0; i<5; i++)
{
 total = in.cieMarks[i] + ec.SemMarks[i];
 System.out.println ("Final marks" + total);
}
}
}
```

Lab Program - 7

```
class Gen<T, S> {
    private T obj;
    private S obj1;
    Gen (T value, S value2) {
        obj = value;
        obj1 = value2;
    }
    T getobj() {
        return obj;
    }
    S get obj () {
        return obj1;
    }

    void objType()
    {
    System.out.println("The type of object" + obj.getclass().
        getName());
    }
    void objType() {
        System.out.println("The type of object" + obj.getclass().
        System.out.println        getName());
```

```
}}
public class Generics {
    public static void main (String [] args)
{
Gen < Integer, Double > ob = new Gen < Integer, Double
    (127, 127.271);

 ob.objType();
System.out.println ("object Value " + ob.getobj());
 ob.objType();

System.out.println ("object value " + obj.getobj());

Gen < String, Integer > ob2 = new Gen < String, Integer >
    ("Ravanth ", 128);

 obj2.objType();
System.out.println ("object value" + obj2.getobj

ob2.objType();
System.out.println ("object value" + obj2.getobj
```

Lab Program - 8

```
import java.util.*;

class ageException extends Exception {
    int detail;
    ageException(int a) {
        detail = a;
    }
    public String toString() {
        return "Exception:" + detail + " the entered age
        does not match";
    }
}
```

~~public String toString()~~
~~return~~

```
class Father
{
    int age;
    Father(int age) throws ageException {
    this.age = age;
    if (this.age <= 0)
    {
    throw new ageException(this.age);
```

```
}}
void display(){
 System.out.println("Father's age:" + this.age);
}}
class Son Extends Father{
 Father f;
Son(int age, Father f) throws age Exception{
 Super(age);
 this.f = f;
if(this.age >= this.f.age)
 {
 throws new age Exception(this.age);
}}

public class DemoExp{
 public static void main(String[] args)
 {
try{
 Scanner input = new Scanner(System.in);
 System.out.print("Enter father's age:");
 Father f = new Father(input.nextInt());
 System.out.print("Enter Son's age:");
 Son s = new Son(input.nextInt(), f);
 }
```

```
catch (Exception e) {
    System.out.println(e);
  }
}
```

Lab Program - 9

```
class Thread implements Runnable {
 Thread t;
 String name;
Thread ( String name)
{
  this.name = name;
    t= new Thread (this, this.name);
     t.start();
}
public void run() {
    try {
     for(int i=0; i<20; i++){
     System.out.println("CSE");
     Thread.sleep(2000);
     }
} catch (InterruptedException e) {
    System.out.println(e);
 }
}
class Thread 2 implement Runnable {
 Thread t;
String name;
 Thread & ( String name) {
```

```
this.name = name;
t = new Thread (this, this.name);
t.start();
}
public void run(){
try{
for (int i=0; i<5; i++){
System.out.println("BMS");
Thread.sleep(10000);
}
} catch (InterruptedException e){
System.out.println(e);
}}}

class DemoThread{
public static void main (String[] args){
Thread1 obj1 = new Thread1("Dept.name");
Thread2 obj2 = new Thread2("College.name");
try{
obj1.t.join();
obj2.t.join();
}
catch (Exception e){
System.out.println("Interrupted");
}}
```

Lab Program-10

```
import java.awt.*;

public class AWTPROG extends Frame
    implements ActionListener{
  TextField t1, t2;
  String msg = " ";
  Button btn;

AWTPROG{
    Label l1 = new Label ("First number:" Label.Right);
  t1 = new TextField(10);

  Label l2 = new Label ("Second number", Label.Right)
t2 = New TextField(10);

btn = new Button ("submit");

   s1.setBackground(Color.Yellow);
   s2.setBackground(Color.Yellow);
    this.add(l1);
     this.add(t1);
     this.add(l2);
      this.add(t2);
```

```
this.add(btn, BorderLayout.ENTER);
this.setVisible(true);
this.setSize(600, 300);
this.setLayout(new FlowLayout
(FlowLayout.center, 20, 10));
btn.addActionListener(this);
addWindowListener(new MyWindow());
```

@Override

```
public void actionPerformed(ActionEvent e)
{
String st1 = t1.getText();
String st2 = t2.getText();
double n1, n2;
n1=0.0;
n2=0.0;
if (st1.equals("") || st2.equals(""))
msg = "You cannot leave the text blank";
else {
try {
n1=Double.parseDouble(st1);
n2=Double.parseDouble(st2);
try{
double res = n1/n2;
```

```
catch (ArithmeticException e1) {
        msg = e1.toString();}

new MyDialog(this, "Result Dialog", false,
  msg, n1, n2);
 }
public static void main(String[] args)
  {
  new AWTPROG1();
  }
  }
class MyDialog extends Dialog implements
   ActionListener {
public MyDialog(Frame owner, String title,
boolean modal, String msg)
{
super(owner, title, modal);
 this.setVisible(true);
 this.setSize(300, 400);
 this.setLayout(new FlowLayout());

Label l1 = new Label("Updates on the result:");
```

```
this.add(l1);
this.add(new Label("Second Number : "+n2));
this.add(new Label(msg));

Button b = new Button("close");
  this.add(b);
  b.addActionListener(this);

@override
  public void actionPerformed(ActionEvent e)
{
 dispose();
}
}
class MyWindow extends WindowAdapter{
  public void windowClosing(WindowEvent e)
  {
  System.exit(0);
  }
  }
```