Part A

1(a) Write a JAVA program to implement class mechanism. –Create a class, methods and invoke them inside main method.

```
class Addition
{
        int sum = 0;

    public int Add(int a, int b)
   {
              sum = a + b;
                  return sum;
    }

}
public class Demo
{
    public static void main (String[] args)
    {
             Addition a = new Addition();
             int s = a.Add(1,2);
            System.out.println("Sum of two integer values :"+ s);

    }
}
```

1(b) Write a JAVA program to implement shift operators in JAVA

```
public class Test
{
   public static void main(String args[])
   {
      int x = -4;
      System.out.println(x>>1);
      int y = 4;
      System.out.println(y>>1);
   }
}
```

2(a) Write a JAVA program to implement constructor 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 Test
{
   public static void main(String args[])
   {
      Box mybox1 = new Box(10, 20, 15);
      Box mybox2 = new Box();
      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);
   }
}
```

2(b) Write a JAVA program to implement for-each loop to compute average of n natural numbers.

```
class ForEach
{
  public static void main(String args[])
{
   int arr[]={12,13,14,44};
   int total=0;
   for(int i:arr)
   {
     total=total+i;
   }
  System.out.println("Total: "+total);
 System.out.println("average="+total  /arr);
 }
}
```

3(a) Write a JAVA program to implement multi level Inheritance

```
class Student
{
    String name = "Raj";
}
class CollegeStudent extends Student
{
    String className = "CSE";
}
class CSE_Student extends CollegeStudent
{
    String semester = "4th";

    public void showDetail()
   {
      System.out.println("Student name = " + name);
      System.out.println("Student class name = " + className);
      System.out.println("Student semester = " + semester);
    }
}
 public class StudentTest
{
    public static void main(String args[])
{
                CSE_Student obj = new CSE_Student();
               obj.showDetail();
    }
}
```

3(b) Write a JAVA program for abstract class to find areas of different shapes.

```
import java.util.Scanner;
 abstract class calcArea
{
   abstract void findTriangle(double b, double h);
   abstract void findRectangle(double l, double b);
   abstract void findSquare(double s);
   abstract void findCircle(double r);
}
 class findArea extends calcArea
{
    void findTriangle(double b, double h)
   {
      double area = (b*h)/2;
      System.out.println("Area of Triangle: "+area);
   }

   void findRectangle(double l, double b)
   {
      double area = l*b;
      System.out.println("Area of Rectangle: "+area);
   }

   void findSquare(double s)
   {
      double area = s*s;
```

```
        System.out.println("Area of Square: "+area);
    }


   void findCircle(double r)
    {
        double area = 3.14*r*r;
        System.out.println("Area of Circle: "+area);
    }
}

class area
{
    public static void main(String args[])
    {
        double l, b, h, r, s;
        findArea area = new findArea();
        Scanner get = new Scanner(System.in);

        System.out.print("\nEnter Base & Vertical Height of Triangle: ");
        b = get.nextDouble();
        h = get.nextDouble();
        area.findTriangle(b, h);

        System.out.print("\nEnter Length & Breadth of Rectangle: ");
        l = get.nextDouble();
        b = get.nextDouble();
        area.findRectangle(l, b);
```

```
        System.out.print("\nEnter Side of a Square: ");
        s = get.nextDouble();
        area.findSquare(s);
        System.out.print("\nEnter Radius of Circle: ");
        r = get.nextDouble();
        area.findCircle(r);
    }
}
```

4(a) Write a JAVA program that describes exception handling mechanism

```
class EceptionHandling
{
        public static void main(String args[])
         {
         try
        {
                    int num1=30, num2=0;
                    int output=num1/num2;
                   System.out.println ("Result: "+output);
        }
        catch(ArithmeticException e)
        {
              System.out.println ("You Shouldn't divide a number by zero");
          }
    }
```

4(b) Write a JAVA program to implement break and continue statements.

```
class BreakStatement
{
   public static void main(String[] args)
   {
      // Initially loop is set to run from 0-9
      for (int i = 0; i < 10; i++)
         {
         // Terminate the loop when i is 5
         if (i == 5)
            break;
         System.out.println("i: " + i);
      }
      System.out.println("Out of Loop");
   }
}

class ContinueStatement
 {
   public static void main(String args[])
   {
      for (int i = 0; i < 10; i++) {
         // If the number is 2
         // skip and continue
         if (i == 2)
            continue;
```

```
            System.out.print(i + " ");
        }
    }
}
```

5a: Write a Java program using IO Streams

```
import java.io.*;
public class CopyFile
{
	public static void main(String args[]) throws IOException
	{
		FileInputStream in = null;
		FileOutputStream out = null;
		try
		{
			in = new FileInputStream("input.txt");
			out = new FileOutputStream("output.txt");
			int c;
			while ((c = in.read()) != -1)
			{
				out.write(c);
			}
		}
		finally
		{
			if (in != null)
			{
				 in.close();
			}
			if (out != null)
			{
				out.close();
			}
		}
	}
}
```

5b: Write a Java program using files

```
import java.io.FileWriter;
public class FileWriterExample
{
	public static void main(String args[])
	 {
		try
		{
			FileWriter fw=new FileWriter("D:\\testout.txt");

			fw.write("Welcome to javaTpoint.");

			fw.close();
		}

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

		System.out.println("Success...");
	}
}
```

Part B

1. **Write a JAVA program that creates threads by extending Thread class .**
   a. **First thread display "Good Morning "every 1 sec,**
   b. **Second thread displays "Hello "every 2 seconds**
   c. **Third display "Welcome" every 3 seconds,(Repeat the same by implementing Runnable.**

```
class A extends Thread
{
    synchronized public void run()
    {
            try
            {
               while(true)
               {
                  sleep(1000);
                  System.out.println("good morning");
               }
            }
            catch(Exception e)
            {

             }
        }
    }
```

```
class B extends Thread
{
    synchronized public void run()
     {
            try
            {
                  while(true)
                  {
                       sleep(2000);
                       System.out.println("hello");
                    }
                 }
                 catch(Exception e)
                  {


                   }
          }
    }
class C extends Thread
{
          synchronized public void run()
          {
                  try
                  {
                       while(true)
                       {
                          sleep(3000);
                          System.out.println("welcome");
                       }
                    }
                    catch(Exception e)
                    {
```

```
                    }
            }
    }
class ThreadDemo
{
   public static void main(String args[])
   {
                    A t1=new A();
                    B t2=new B();
                    C t3=new C();
                    t1.start();
                    t2.start();
                    t3.start();
    }
}
```

2. WAP to implement ProducerConsumer problem:

```
public class ProducerConsumerTest
{
  public static void main(String[] args)
  {
    Queue q = new Queue( );
    Producer p1 = new Producer(q, 1);
    Consumer c1 = new Consumer(q, 1);
    p1.start( );
    c1.start( );
  }
}
class Queue
{
  private int contents;
  private boolean available = false;
  public synchronized int get( )
  {
    while (available == false)
    {
      try
      {
        wait();
      }
      catch (InterruptedException e) {}
    }
    available = false;
```

```
    notifyAll( );
    return contents;
  }

  public synchronized void put(int value)
  {
    while (available == true)
    {
      try
      {
        wait();
      }
      catch (InterruptedException e) { }
    }
    contents = value;
    available = true;
    notifyAll();
  }
}
class Consumer extends Thread
{
  private Queue q;
  private int number;
  public Consumer(Queue c, int number)
  {
    q= c;
    this.number = number;
  }
```

```
  public void run()
  {
    int value = 0;
    for (int i = 0; i < 10; i++)
    {
      value = q.get();
      System.out.println("Consumer #" + this.number + " got: " + value);
    }
  }
}
class Producer extends Thread
{
  private Queue q;
  private int number;

  public Producer(Queue c, int number)
  {
    q= c;
    this.number = number;
  }
  public void run()
  {
    for (int i = 0; i < 10; i++)
    {
      q.put(i);
      System.out.println("Producer #" + this.number + " put: " + i);
      try
      {
```

```
            Thread.sleep(2000);
        }
        catch (InterruptedException e) { }
      }
   }
}
```

3. **Write a JAVA program to create an applet and set its background color and foreground color displaying a message**

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
 public class SetBackColor extends Applet
{
 public void init()
 {
  setBackground(Color.cyan);
  setForeground(Color.red);
 }
  public void paint(Graphics g)
 {
  g.drawString("Hello Java",50,50);
 }
}
/*
<applet code="SetBackColor" width=200 height=200>
</applet>
*/
```

4. Write a Java program to demonstrate key event handlers using deligation event model

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
 <applet code="SimpleKey" width=500 height=300>
 </applet>
*/
  public class SimpleKey extends Applet implements KeyListener
{
    String msg = "";
    int X = 10, Y = 20;

    public void init()
    {
       addKeyListener(this);
    }
     public void keyPressed(KeyEvent ke)
    {
       showStatus("Key Down");
    }
      public void keyReleased(KeyEvent ke)
    {
       showStatus("Key Up");
    }
```

```
public void keyTyped(KeyEvent ke)
   {
      msg += ke.getKeyChar();
      repaint();
   }

public void paint(Graphics g)
   {
      g.drawString(msg, X, Y);
   }
}
```