



**University of Engineering & Management, Kolkata**

**Department of Computer Science & Engineering**

**Course: B.Tech. CSE / CSE (AIML) / CSE (IOT-CYS-BCT) / CSBS**

**Semester: 6th**

**Paper Name: Object Oriented Programming using JAVA Laboratory**

**Paper Code: PCCCSE693**

## Assignment 1

### 1. Write a Java Program to Print "HELLO JAVA".

```
class program1
{
    public static void main(String args[])
    {
        System.out.println("Hello Java");
    }
}
```

### 2. Write a Java Program to add two numbers by declaring variables value.

```
class program2
{
    public static void main(String args[])
    {
        int a=5,b=6,c;
        c=a+b;
        System.out.println("Sum = "+c);
    }
}
```

### 3. Write a Java Program to calculate factorial value of a declared variable.

```
class program3
{
    public static void main(String args[])
    {
        int a=5,f=1;
        for(int i=1; i<=a;i++)
        {
            f=f*i;
        }
        System.out.println("Factorial of "+a+" = "+f);
    }
}
```

### 4. Write a Java Program to calculate factorial value of a declared variable by creating separate method for factorial segment.

```
import java.util.*;
class program4
{
    static int factorial(int a)
    {
        int f=1;
        for(int i=1; i<=a;i++)
        {
            f=f*i;
        }
        return f;
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        int a=5;
        program4 obj = new program4();
        System.out.println("Factorial of "+a+" = "+ obj.factorial(a));
    }
}
```

**5. Write a Java Program to calculate factorial value of a declared variable by creating separate class and method for factorial segment.**

```
import java.util.*;
class fact
{
    int factorial(int a)
    {
        int f=1;
        for(int i=1; i<=a;i++)
        {
            f=f*i;
        }
        return f;
    }
}

class program5
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        int a=5;
        fact obj = new fact();
        System.out.println("Factorial of "+a+" = "+ obj.factorial(a));
    }
}
```

**6. Write a Java Program to calculate factorial value of a variable by taking input from command line.**

```
import java.util.*;
class program6
{
    public static void main(String[] args)
    {
        int f=1;
        for(int i=1; i<=Integer.parseInt(args[0]);i++)
        {
            f=f*i;
        }
        System.out.println("Factorial = "+f);
    }
}
```

**7. Write a Java Program to display whether a number is odd or even.**

```
import java.util.*;
class program7
{
    public static void main(String args[])
    {
        if(Integer.parseInt(args[0])%2!=0)
            System.out.println("Odd");
        else
            System.out.println("Even");
    }
}
```

**8. Write a Java program to find maximum of three numbers.**

```
import java.util.*;
class program8
{
    public static void main(String args[])
    {
        int
a=Integer.parseInt(args[0]),b=Integer.parseInt(args[1]),c=Integer.parseInt(args[2]);

        System.out.println("Max = "+((a>b)?(a>c)?a:c:(b>c)?b:c));
    }
}
```

**9. Write a Java program to swap two numbers.**

```
import java.util.*;
class program9
{
    public static void main(String args[])
    {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        a=a^b;
        b=a^b;
        a=a^b;
        System.out.println("a = "+a);
        System.out.println("b = "+b);
    }
}
```

**10. Write a Java program to check whether a year is leap year or not.**

```
import java.util.*;
class program10
{
    public static void main(String args[])
    {
        int y = Integer.parseInt(args[0]);
        if(y%400==0 || (y%100!=0 && y%4==0))
            System.out.println("Leap Year");
        else
            System.out.println("Non Leap Year");
    }
}
```

**11. Write a Java program for following grading system.**

**Note: Percentage >= 90% : Grade A**

**Percentage >= 80% : Grade B**

**Percentage >= 70% : Grade C**

**Percentage >= 60% : Grade D**

**Percentage >= 40% : Grade E**

**Percentage < 40% : Grade F**

```
import java.util.*;
class program11
{
    public static void main(String args[])
    {
        int m = Integer.parseInt(args[0]);
        if(m >= 90)
            System.out.println("Grade A");
        else if(m >= 80)
            System.out.println("Grade B");
        else if(m >= 70)
            System.out.println("Grade C");
        else if(m >= 60)
            System.out.println("Grade D");
        else if(m >= 40)
            System.out.println("Grade E");
        else
            System.out.println("Grade F");
    }
}
```

**12. Write a Java program to check whether a number is divisible by 5 or not.**

```
import java.util.*;
class program12
{
    public static void main(String args[])
    {
        int m = Integer.parseInt(args[0]);
        if(m % 5 == 0)
            System.out.println("Divisible by 5");
        else
            System.out.println("Not Divisible by 5");
    }
}
```

## Assignment 2

### 1. Write a java program to create a simple array and access array element.

```
import java.util.*;
class program1
{
    public static void main(String args[])
    {
        int arr[]={1,2,3,4,5};
        System.out.println(arr[2]);
    }
}
```

### 2. Write a java program to create 2D array and access the array element.

```
import java.util.*;
class program2
{
    public static void main(String args[])
    {
        int arr[][]={{1,2},{3,4},{5,6}};
        System.out.println(arr[0][0]);
    }
}
```

### 3. Write a Java program to find the sum of even numbers in an integer array.

```
import java.util.*;
class program3
{
    public static void main(String args[])
    {
        int arr[]={1,2,3,4,5,6};
        for(int i=0;i<arr.length;i++)
        {
            if (arr[i]%2==0)
                System.out.println(arr[i]);
        }
    }
}
```

### 4. Write a Java program to calculate Sum of two 2-dimensional arrays.

```
class program4
{
    public static void main(String args[])
    {
        int arr1[][]={{1,2,3},{4,5,6}};
        int arr2[][]={{6,7,8},{9,10,11}};
        for(int i=0;i<arr1.length;i++)
        {
            for(int j=0;j<arr1[0].length;j++)
            {
                System.out.print(arr1[i][j]+arr2[i][j]+"\\t");
            }
            System.out.println();
        }
    }
}
```

**5. Write a Java program to find the sum of diagonal elements in a 2D array.**

```
class program5
{
    public static void main(String args[])
    {
        int arr1[][]={{1,2,3},{4,5,6},{9,10,11}},s=0;
        for(int i=0;i<arr1.length;i++)
        {
            for(int j=0;j<arr1[0].length;j++)
            {
                if (i==j)
                    s+=arr1[i][j];
            }
        }
        System.out.println("Sum Of diagonal = "+s);
    }
}
```

**6. Write a Java program to multiply two matrices.**

```
class program6
{
    public static void main(String args[])
    {
        int arr1[][]={{1,2,3},{4,5,6},{9,10,11}};
        int arr2[][]={{6,7,8},{9,10,11},{1,2,3}};
        int a[][]= new int[arr1[0].length][arr2.length];
        for(int i=0;i<arr1.length;i++)
            for(int k=0;k<arr2.length;k++)
                for(int j=0;j<arr1[0].length;j++)
                    a[i][k]+=arr1[i][j] * arr2[j][k];
        System.out.println("Matrix Multiplication");
        for(int i=0;i<arr1.length;i++)
        {
            for(int k=0;k<arr2.length;k++)
                System.out.print(a[i][k]+"\\t");
            System.out.println();
        }
    }
}
```

**7. Implementation of default, no argument Constructor.**

```
class program7
{
    int x;
    program7()
    {
        x=10;
    }
    public static void main(String args[])
    {
        program7 ob = new program7();
        System.out.println(ob.x);
    }
}
```

**8. Implementation of parameterized constructor.**

```
class program8
{
    int x;
    public static void main(String args[])
    {
        program8 ob = new program8();
        System.out.println(ob.x);
    }
}
```

### 9. Implementation of returning the value from the caller method.

```
class program9
{
    int sum(int a, int b){return(a+b);}
    public static void main(String args[])
    {
        program9 ob = new program9();
        System.out.println("Sum= "+ob.sum(5,6));
    }
}
```

### 10. Implementation of call by value and call by reference.

```
class program10
{
    int x;
    void func(program10 p){
        p.x=20;
    }
    public static void main(String args[])
    {
        program10 ob = new program10();
        ob.x=10;
        ob.func(ob);
        System.out.println("Call by reference example x= "+ob.x);
    }
}
```



## Assignment 3

### 1. Write a java program to implement BufferedReader class.

```
import java.io.*;
public class program1
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader reader =new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter a number: ");
        int n = Integer.parseInt(reader.readLine());
        System.out.println("Your number:"+n);
    }
}
```

### 2. Write a java program to take input from keyboard using Scanner class.

```
import java.util.*;
class program2
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = in.nextInt();
        System.out.println("Your number:"+n);
    }
}
```

### 3. Write a Java program to reverse a number.

```
import java.util.*;
class program3
{
    int reverse(int n)
    {
        int rev=0;
        while(n>0)
        {
            rev=rev*10+n%10;
            n/=10;
        }
        return(rev);
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        program3 ob = new program3();
        System.out.print("Enter a number: ");
        System.out.println("Reverse number: "+ob.reverse(in.nextInt()));
    }
}
```

### 4. Write a Java program to check whether a number is palindrome or not.

```
import java.util.*;
class program4
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        program3 ob = new program3();
        System.out.print("Enter a number: ");
        int n=in.nextInt();
        System.out.println((ob.reverse(n)==n)?"Palindrome":"Not Plaindrome");
    }
}
```

**5. Write a Java program to check whether a number is prime or not.**

```
import java.util.*;
class program5
{
    boolean prime(int i,int n)
    {
        if(i==n)
            return true;
        else if(n%i==0 || n==1)
            return false;
        return prime(i+1,n);
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        program5 ob = new program5();
        System.out.print("Enter a number: ");
        System.out.println((ob.prime(2,in.nextInt()))?"Prime":"Not Prime");
    }
}
```

**6. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.**

```
import java.util.*;
class program6
{
    String decimalToBinary(int n)
    {
        String s="";
        do
        {
            s=Integer.toString(n%2)+s;
            n/=2;
        }while(n>0);
        return(s);
    }
    int binaryToDecimal(int n)
    {
        int x=0,c=0;
        do
        {
            x+=n%10*Math.pow(2,c++);
            n/=10;
        }while(n>0);
        return(x);
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        program6 ob = new program6();
        System.out.print("Enter a number: ");
        System.out.println("Binary representation: "+ob.decimalToBinary(in.nextInt()));
        System.out.print("Enter a binary number: ");
        System.out.println("Decimal representation: "+ob.binaryToDecimal(in.nextInt()));
    }
}
```

**7. Write a Java program to check whether a given number is Armstrong Number or not.**

```
import java.util.*;
class program7
{
    boolean armstrong(int n)
    {
        int c=n,x=0,l=(int)Math.log10(n)+1;
        while(n>0)
        {
            x+=Math.pow(n%10,l);
        }
    }
}
```

```

        n/=10;
    }
    return(x==c);
}
public static void main(String args[])
{
    Scanner in = new Scanner(System.in);
    program7 ob = new program7();
    System.out.print("Enter a number: ");
    if(ob.armstrong(in.nextInt()))
        System.out.println("Armstrong Number");
    else
        System.out.println("Not Armstrong Number");
}
}

```

#### 8. Write a Java program to calculate the sum of natural numbers up to a certain range.

```

import java.util.*;
class program8
{
    int x;
    public static void main(String args[])
    {
        program8 ob = new program8();
        Scanner in = new Scanner(System.in);
        System.out.print("Enter the last number of the range: ");
        int n=in.nextInt();
        if (n<1)
            System.out.println("Invalid Input");
        else
            System.out.println("Sum of natural till the range: "+(n*(n+1))/2);
    }
}

```

#### 9. Write java codes to implement the followings –

**Basic string handling concepts- Concept of mutable and immutable string, Methods of String class-charAt(), compareTo(), equals(), equalsIgnoreCase(), indexOf(), length() , substring().; toCharArray(), toLowerCase(), toString(), toUpperCase() , trim() , valueOf() methods,**

```

import java.util.*;
class program9
{
    public static void main(String args[])
    {
        String s=" Hello World ";
        System.out.println("Character at third place: "+s.charAt(2));
        System.out.println("Compare to: "+s.compareTo("Java"));
        System.out.println("Equals : "+s.equals(" hello world "));
        System.out.println("Equals Ignorecase : "+s.equalsIgnoreCase(" hello world "));
        System.out.println("Index of H : "+s.indexOf('H'));
        System.out.println("Length : "+s.length());
        System.out.println("Substring : "+s.substring(0,5));
        System.out.print("Char Array : ");
        char ch[]=s.toCharArray();
        for(int i=0;i<s.length();i+=2)
        {
            System.out.print(s.charAt(i));
        }
        System.out.println("\nLower Case : "+s.toLowerCase());
        System.out.println("Upper Case : "+s.toUpperCase());
        System.out.println("Integer to String : "+Integer.toString(12));
        System.out.println("Trim : "+s.trim());
        boolean bol = true;
        System.out.println("Value of : "+String.valueOf(bol));
    }
}

```

**10. Write java codes to implement the followings –**

**Methods of StringBuffer class: append(), capacity(), charAt(), delete(), deleteCharAt().;**

**ensureCapacity(), getChars(), indexOf(), insert(), length(), setCharAt(), setLength(), substring(), toString().**

```
class program10
{
    public static void main(String args[])
    {
        program10 ob = new program10();
        StringBuffer sb = new StringBuffer("Hello ");
        sb.append("Java "); // now original string is changed
        System.out.println("Append: "+sb);
        sb.insert(6, "World ");
        System.out.println("Insert: "+sb);
        sb.deleteCharAt(5);
        System.out.println("DeleteCharAt: "+sb);
        sb.replace(1,5,"i ");
        System.out.println("Replace: "+sb);
        sb.delete(0,3);
        System.out.println("Delete: "+sb);
        System.out.println("Capacity: "+sb.capacity());
        sb.setCharAt(10,'!');
        System.out.println("SetCharAt: "+sb);
        sb.reverse();
        System.out.println("Reverse: "+sb);
    }
}
```

## Assignment 4

### **1. Implementation of method overloading with respect to parameter number, parameter data type.**

```
class program1
{
    void f1(int a)
    {
        System.out.println("from f1(int a)");
    }
    void f1(int a,int b)
    {
        System.out.println("from f1(int a,int b)");
    }
    void f1(float a)
    {
        System.out.println("from f1(float a)");
    }
    public static void main(String args[])
    {
        program1 ob = new program1();
        ob.f1(10);
        ob.f1(10,20);
        ob.f1(10.0f);
    }
}
```

### **2. Write a java program to implement of constructor overloading.**

```
class program2
{
    program2(int a)
    {
        System.out.println("Constructor(int a)");
    }
    program2(int a,int b)
    {
        System.out.println("Constructor(int a,int b)");
    }
    program2(float a)
    {
        System.out.println("Constructor(float a)");
    }
    public static void main(String args[])
    {
        program2 ob1 = new program2(10);
        program2 ob2 = new program2(10,20);
        program2 ob3 = new program2(10.0f);
    }
}
```

### **3. Implementation of this keyword to invoke current class method.**

```
class program3
{
    int a;
    program3(int a)
    {
        this.a=a;
    }
    public static void main(String args[])
    {
        program3 ob1 = new program3(10);
        System.out.println(ob1.a);
    }
}
```

#### 4. Implementation of this keyword to invoke current class constructor.

```
class program4
{
    int a;
    program4(int a)
    {
        this.a=a;
    }
    program4()
    {
        this(10);
    }
    public static void main(String args[])
    {
        program4 ob1 = new program4();
        System.out.println(ob1.a);
    }
}
```

#### 5. Implementation of this keyword to pass as an argument in the method.

```
class program5
{
    int a;
    void increment(int a)
    {
        this.a+=a;
        print(this.a);
    }
    void print(int x)
    {
        System.out.println(x);
    }
    public static void main(String args[])
    {
        program5 ob = new program5();
        ob.increment(10);
    }
}
```

#### 6. Implementation of this keyword to pass as argument in the constructor call.

```
class program6
{
    int a;
    program6()
    {
        a=60;
        program6 obj = new program6(this);
        System.out.println(obj.a);
    }
    program6(program6 obj)
    {
        this.a=obj.a+10;
    }
    public static void main(String args[])
    {
        program6 ob = new program6();
    }
}
```

#### 7. Implement - this keyword can be used to return current class instance.

```
class program7
{
    int a;
    program7 increment(int a)
    {
        this.a=a;
```

```

        return this;
    }
    void print()
    {
        System.out.println(a);
    }
    public static void main(String args[])
    {
        new program7().increment(30).print();
    }
}

```

**8. Prove that this keyword refers to the current class instance variable.**

```

class program8
{
    program8()
    {
        System.out.println(this);
    }
    public static void main(String args[])
    {
        program8 ob = new program8();
        System.out.println(ob);
    }
}

```

**9. Implementation of this keyword as local variable suppressor.**

```

class program9
{
    int a;
    program9(int a)
    {
        System.out.println("a: "+this.a);
    }
    public static void main(String args[])
    {
        program9 ob = new program9(10);
    }
}

```

**10. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume ( ). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.**

```

import java.util.*;

class ThreeDObject {

    Scanner in = new Scanner(System.in);

    double wholeSurfaceArea() {
        return (0.0);
    }

    double volume() {
        return (0.0);
    }
}

class Box extends ThreeDObject {

    int l, b, h;

```

```

Box() {
    System.out.println("Enter length, breadth and height of the box: ");
    l = in.nextInt();
    b = in.nextInt();
    h = in.nextInt();
}

double wholeSurfaceArea() {
    return (2.0 * (l * b + b * h + l * h));
}

double volume() {
    return (l * b * h);
}
}

class Cube extends ThreeDObject {

    int s;

    Cube() {
        System.out.println("Enter side of the cube: ");
        s = in.nextInt();
    }

    double wholeSurfaceArea() {
        return (6.0 * s * s);
    }

    double volume() {
        return (s * s * s);
    }
}

class Cylinder extends ThreeDObject {

    int r, h;

    Cylinder() {
        System.out.println("Enter radius and height of the cylinder: ");
        r = in.nextInt();
        h = in.nextInt();
    }

    double wholeSurfaceArea() {
        return (2.0 * Math.PI * r * (r + h));
    }

    double volume() {
        return (Math.PI * r * r * h);
    }
}

class Cone extends ThreeDObject {

    int r, h;

    Cone() {
        System.out.println("Enter radius and height of the cylinder: ");
        r = in.nextInt();
        h = in.nextInt();
    }

    double wholeSurfaceArea() {
        return (Math.PI * r * (r + Math.sqrt(h * h + r * r)));
    }
}

```



```

    double volume() {
        return (1 / 3.0 * Math.PI * r * r * h);
    }
}

class program10 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println(
            "3D Object Calculations\n1>Box\n2>Cube\n3>Cylinder\n4>Cone\nEnter Your Choice:"
        );
        ThreeDObject ob;
        int ch = in.nextInt();

        switch (ch) {
            case 1:
                ob = new Box();
                break;
            case 2:
                ob = new Cube();
                break;
            case 3:
                ob = new Cylinder();
                break;
            case 4:
                ob = new Cone();
                break;
            default:
                ob = new ThreeDObject();
        }
        System.out.println("Surface Area: " + ob.wholeSurfaceArea());
        System.out.println("Volume: " + ob.volume());
    }
}

```

## Assignment 5

### 1. Write a java program to implement the static keyword in java.

```
import java.io.*;
public class program1
{
    static int x;
    void increment()
    {
        x+=5;
    }
    public static void main(String args[])
    {
        program1 ob1 = new program1();
        ob1.increment();
        System.out.println(ob1.x);
        program1 ob2 = new program1();
        ob2.increment();
        System.out.println(ob2.x);
    }
}
```

### 2. Write a java program to implement the static method in java.

```
import java.util.*;
public class program2
{
    static int x;
    static void increment()
    {
        x+=5;
    }
    public static void main(String args[])
    {
        increment();
        System.out.println(x);
        increment();
        System.out.println(x);
    }
}
```

### 3. Write a java program to implement the single inheritance in java.

```
import java.util.*;
class A
{
    int x;
    void increment()
    {
        x+=5;
    }
}
public class program3 extends A
{
    public static void main(String args[])
    {
        program3 ob = new program3();
        ob.increment();
        System.out.println(ob.x);
    }
}
```

**4. Write a java program to implement the hierarchical inheritance in java.**

```
class A
{
    int x;
    A()
    {x=0;}
    void initialize()
    {
        x=5;
    }
}
class B extends A
{
    B()
    {super();}
    void print()
    {
        System.out.println("From B: "+x);
    }
}
class C extends A
{
    C()
    {super();}
    void print()
    {
        System.out.println("From C: "+x);
    }
}
public class program4
{
    public static void main(String args[])
    {
        B ob1 = new B();
        ob1.initialize();
        ob1.print();
        C ob2 = new C();
        ob2.initialize();
        ob2.print();
    }
}
```

**5. Write a java program to implement the multilevel inheritance in java.**

```
import java.io.*;
class A
{
    int x;
    void increment()
    {
        x+=5;
    }
}
class B extends A
{
    void decrement()
    {
        x-=2;
    }
}
class C extends B
{
    void print()
    {
        System.out.println("From C: "+x);
    }
}
```

```

public class program5
{
    public static void main(String args[])
    {
        C ob = new C();
        ob.increment();
        ob.decrement();
        ob.print();
    }
}

```

## 6. Multiple inheritance does not support in java – justify.

```

import java.util.*;
class program6
{
    String decimalToBinary(int n)
    {
        String s="";
        do
        {
            s=Integer.toString(n%2)+s;
            n/=2;
        }while(n>0);
        return(s);
    }
    int binaryToDecimal(int n)
    {
        int x=0,c=0;
        do
        {
            x+=n%10*Math.pow(2,c++);
            n/=10;
        }while(n>0);
        return(x);
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        program6 ob = new program6();
        System.out.print("Enter a number: ");
        System.out.println("Binary representation: "+ob.decimalToBinary(in.nextInt()));
        System.out.print("Enter a binary number: ");
        System.out.println("Decimal representation: "+ob.binaryToDecimal(in.nextInt()));
    }
}

```

## 7. Implementation of method overriding in java.

```

import java.io.*;
class A
{
    int x;
    void increment()
    {
        System.out.println("Called A");
        x+=5;
    }
}
class B extends A
{
    void increment()
    {
        System.out.println("Called B");
        x+=10;
    }
}
class C extends B

```

```

{
    void print()
    {
        System.out.println("From C: "+x);
    }
}
public class program7
{
    public static void main(String args[])
    {
        C ob = new C();
        ob.increment();
        ob.print();
    }
}

```

### 8. Implementation of dynamic method dispatch in java.

```

import java.io.*;
class A
{
    int x;
    void increment()
    {
        System.out.println("Called A");
        x+=5;
    }
}
class B extends A
{
    void increment()
    {
        System.out.println("Called B");
        x+=10;
    }
}
public class program8
{
    public static void main(String args[])
    {
        B ob1 = new B();
        A ob2 = new A();
        A ob3;
        if (Integer.parseInt(args[0])==1)
            ob3=ob1;
        else
            ob3=ob2;
        ob3.increment();
        System.out.println(ob3.x);
    }
}

```

### 9. Write a java program to stop method overriding.

```

import java.io.*;
class A
{
    int x;
    final void increment()
    {
        System.out.println("Called A");
        x+=5;
    }
}
class B extends A
{
    void increment()
    {

```

```

        System.out.println("Called B");
        x+=10;
    }
}
public class program9
{
    public static void main(String args[])
    {
        B ob1 = new B();
        A ob2 = new A();
        A ob3;
        if (Integer.parseInt(args[0])==1)
            ob3=ob1;
        else
            ob3=ob2;
        ob3.increment();
        System.out.println(ob3.x);
    }
}

```

**10. Create a “circle” class & a “point” class. The coordinates of the circle are given and used within the “circle” class as object of the “point” class. Display the area of circle.**

```

import java.util.*;
class Point {

    int x;
    int y;

    Point(){
        x = y = 0;
    }

    Point(int x, int y){
        this.x = x;
        this.y = y;
    }

    double distance (Point ob){
        return(Math.sqrt(Math.pow(this.x-ob.x,2)+Math.pow(this.y-ob.y,2)));
    }
}

class Circle extends Point{
    double radius;

    void calculate_radius()
    {
        Scanner in = new Scanner(System.in);
        Point ob = new Point();
        System.out.println("Enter a coordinate: ");
        ob.x=in.nextInt();
        ob.y=in.nextInt();
        radius=distance(ob);
    }

    void area() {
        System.out.println("Area of Circle: "+Math.PI * radius * radius);
    }
}

class program10
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
    }
}

```

```

        Circle obj = new Circle();
        System.out.println("Enter a coordinate: ");
        obj.x=in.nextInt();
        obj.y=in.nextInt();
        obj.calculate_radius();
        obj.area();
    }
}

```

**11. Write a program to define a class Employee to accept emp\_id, emp\_name, basic\_salary from the user and display the gross\_salary.**

```

import java.util.*;

class Employee {

    int emp_id;
    String emp_name;
    double basic_salary;

    void display() {

        double da=basic_salary*15/100;
        double hra=basic_salary*10/100;
        double gross_sal=basic_salary+da+hra;
        System.out.println ("Employee Id= "+emp_id);
        System.out.println ("Employee Name= "+emp_name);
        System.out.println ("Gross Salary= "+gross_sal);
    }
}

class program11
{
    public static void main(String[] args){

        Scanner in = new Scanner(System.in);
        Employee ob = new Employee();
        System.out.println ("Enter Employee id");
        ob.emp_id = in.nextInt();
        System.out.println ("Enter Employee Name");
        in.nextLine();
        ob.emp_name = in.nextLine();
        System.out.println ("Enter Basic Salary");
        ob.basic_salary = in.nextDouble();
        ob.display();
    }
}

```

**12. Write a program to define a class Fraction having data members numerator and denominator. Initialize three objects using different constructors and display its fractional value.**

```

class Fraction {

    int numerator,denominator;

    Fraction () {

        numerator=0;
        denominator=1;
    }

    Fraction (int x, int y) {

        numerator=x;
        denominator=y;
    }
}

```

```

Fraction(int x) {
    numerator=x;
    denominator=1;
}

void display() {
    System.out.println ("Fraction = "+(double)(numerator)/denominator);
}
}

class program12
{
    public static void main(String[] args) {
        Fraction f1 = new Fraction();
        f1.display();
        Fraction f2 = new Fraction(10,12);
        f2.display();
        Fraction f3 = new Fraction(15);
        f3.display();
    }
}

```

**13. Write a program to define a class Item containing code and price. Accept this data for five objects using array of objects. Display code, price in tabular form and also, display total price of all items.**

```

import java.util.*;

class Item {
    String code;
    int price;
}

class program13
{
    public static void main(String args[])
    {
        Item[] I = new Item[5];
        int total_cost=0;
        Scanner in = new Scanner(System.in);
        for(int i=0;i<5;i++) {
            I[i] = new Item();
            System.out.print("Enter code for item"+(i+1)+" : ");
            I[i].code = in.next();
            System.out.print("Enter price for item"+(i+1)+" : ");
            I[i].price = in.nextInt();

        }

        System.out.println("Item Code Item Price");
        System.out.println("-----");
        for(int i=0;i<5;i++)
        {
            System.out.print("\t"+I[i].code);
            System.out.println("\t\t"+I[i].price);
            total_cost = total_cost + I[i].price;
        }
        System.out.print("Total Price = "+total_cost);
    }
}

```



## Assignment 6

### 1. Write a java program to implement the abstraction property.

```
abstract class A
{
    abstract int sum(int a, int b);
}
class B extends A
{
    int sum(int a, int b)
    {
        return a+b;
    }
}
public class program1
{
    public static void main(String args[])
    {
        B ob = new B();
        System.out.println(ob.sum(5,6));
    }
}
```

### 2. Write a java program to implement interface.

```
interface A
{
    abstract int sum(int a, int b);
}
class B implements A
{
    public int sum(int a, int b)
    {
        return a+b;
    }
}
public class program2
{
    public static void main(String args[])
    {
        B ob = new B();
        System.out.println(ob.sum(5,6));
    }
}
```

### 3. Write a java program to implement multi level inheritance with the help of interface.

```
interface A
{
    abstract int sum(int a, int b);
}
interface B extends A
{
    abstract void print(int sum);
}
class C implements B
{
    public int sum(int a, int b)
    {
        return a+b;
    }
    public void print(int sum)
    {
        System.out.println("Sum= "+sum);
    }
}
public class program3
```

```

{
    public static void main(String args[])
    {
        C ob = new C();
        ob.print(ob.sum(5,6));
    }
}

```

#### 4. Write a java program to implement the inheritance in interface.

```

interface A
{
    abstract int sum(int a, int b);
}
interface B extends A
{
    abstract void print(int sum);
}
class C implements B
{
    public int sum(int a, int b)
    {
        return a+b;
    }
    public void print(int sum)
    {
        System.out.println("Sum= "+sum);
    }
}
public class program4
{
    public static void main(String args[])
    {
        C ob = new C();
        ob.print(ob.sum(5,6));
    }
}

```

#### 5. Write a java program to implement multiple inheritance using interface.

```

interface A
{
    abstract int sum(int a, int b);
}
interface B
{
    abstract void print(int sum);
}
class C implements A,B
{
    public int sum(int a, int b)
    {
        return a+b;
    }
    public void print(int sum)
    {
        System.out.println("Sum= "+sum);
    }
}
public class program5
{
    public static void main(String args[])
    {
        C ob = new C();
        ob.print(ob.sum(5,6));
    }
}

```

**6. Write a java program to implement super keyword in java.**

```
class A
{
    int sum(int a, int b)
    {
        return a+b;
    }
}
class B extends A
{
    void print(int a, int b)
    {
        System.out.println("Sum= "+super.sum(a,b));
    }
}
public class program6
{
    public static void main(String args[])
    {
        B ob = new B();
        ob.print(5,6);
    }
}
```

**7. Write a java program to implement super() method without parameter.**

```
abstract class A
{
    A()
    {
        System.out.println("Constructor of A");
    }
}
class B extends A
{
    B()
    {
        super();
        System.out.println("Constructor of B");
    }
}
public class program7
{
    public static void main(String args[])
    {
        B ob = new B();
    }
}
```

**8. Write a java program to implement super() method with parameter.**

```
class A
{
    A(int a)
    {
        System.out.println("Constructor of A: "+a);
    }
}
class B extends A
{
    B(int a,int b)
    {
        super(a);
        System.out.println("Constructor of B: "+b);
    }
}
public class program8
{
}
```

```

    public static void main(String args[])
    {
        B ob = new B(5,6);
    }
}

```

**9. Create an interface called Player. The interface has an abstract method called play() that displays a message describing the meaning of “play” to the class. Create classes called Child, Musician, and Actor that all implement Player. Create an application that demonstrates the use of the classes(UsePlayer.java**

```

interface Player
{
    abstract void play();
}
class Child implements Player
{
    public void play()
    {
        System.out.println("Child plays with toys");
    }
}
class Musician implements Player
{
    public void play()
    {
        System.out.println("Musician plays music");
    }
}
class Actor implements Player
{
    public void play()
    {
        System.out.println("Actor plays his/her role");
    }
}
public class program9
{
    public static void main(String args[])
    {
        Child ob1 = new Child();
        ob1.play();
        Musician ob2 = new Musician();
        ob2.play();
        Actor ob3 = new Actor();
        ob3.play();
    }
}

```

**10. Create an abstract class Accounts with the following details:**

**Data Members:** Balance (b) accountNumber (c) accountHoldersName (d) address

**Methods:** withdrawl()- abstract , (b) deposit()- abstract , (c) display() to show the balance of the account number

**Create a subclass of this class SavingsAccount and add the following details:**

**Data Members:** (a) rateOfInterest

**Methods:** (a) calculateAount()

```

import java.util.*;

abstract class Accounts {

    double balance;
    String accountNumber, accountHoldersName, address;

    Accounts(double bal, String an, String name, String add) {

```

```

        balance = bal;
        accountNumber = an;
        accountHoldersName = name;
        address = add;
    }

    abstract void withdrawl(double amt);

    abstract void deposit(double amt);

    void display() {
        System.out.println(
            "Balance of Account No( " + accountNumber + " ) = " + balance
        );
    }
}

class SavingsAccount extends Accounts {

    double rateOfInterest;

    SavingsAccount(double bal, String an, String name, String add, double rate) {
        super(bal, an, name, add);
        rateOfInterest = rate;
    }

    public void withdrawl(double amt) {
        if (amt <= balance) {
            System.out.println("Withdrawal Successful");
            balance -= amt;
            display();
        } else {
            System.out.println("Requested amount is greater than Balance");
            display();
        }
    }

    public void deposit(double amt) {
        System.out.println("Withdrawal Successful");
        balance += amt;
        display();
    }

    void calculateAmount() {
        double amt = (balance + balance * rateOfInterest / 100.0);
        System.out.println("Amount = " + amt);
        balance = amt;
        System.out.println("Balance Updated");
        display();
    }
}

public class program10 {

    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        double bal, rate;
        String an, name, add;
        System.out.println("Enter Account Number: ");
        an = in.next();
        in.nextLine();
        System.out.println("Enter Account Holder Name: ");
        name = in.nextLine();
        System.out.println("Enter Address: ");
        add = in.nextLine();
        System.out.println("Enter Starting Balance: ");
        bal = in.nextDouble();
    }
}

```

```

System.out.println("Enter Rate Of Interest: ");
rate = in.nextDouble();
SavingsAccount ob = new SavingsAccount(bal, an, name, add, rate);
System.out.println("Bank Account Successfully Created");
int ch;
w:while (true) {
    System.out.println(
        "\n\n*****"
    );
    System.out.println("Bank Menu");
    System.out.println("1. Withdrawal");
    System.out.println("2. Deposit");
    System.out.println("3. Calculate Amount");
    System.out.println("4. Exit");
    System.out.println("Enter your choice: ");
    ch = in.nextInt();
    double amt;
    switch (ch) {
        case 1:
            System.out.println("Enter Withdrawal Amount : ");
            amt = in.nextDouble();
            ob.withdrawl(amt);
            break;
        case 2:
            System.out.println("Enter Deposit Amount : ");
            amt = in.nextDouble();
            ob.deposit(amt);
            break;
        case 3:
            ob.calculateAmount();
            break;
        case 4:
            System.out.println("Thank You");
            break w;
        default:
            System.out.println("Invalid Option");
    }
}
}
}

```

## Assignment 7

### 1. Implementation of final keyword before a variable.

```
class program1
{
    public static void main(String args[])
    {
        final double pi = 3.14;
        pi=3.1432;
    }
}
```

### 2. Implementation of final keyword before a method.

```
class A
{
    final void f1()
    {
        System.out.println("From A");
    }
}
class B extends A
{
    void f1()
    {
        System.out.println("From B");
    }
}
```

### 3. Implementation of final keyword before a class.

```
final class A
{
}
class B extends A
{
}
```

### 4. Write a java program to create package.

```
package p1;
public class program4
{
    public static int sum(int a, int b)
    {
        return a + b;
    }
    static int diff(int a, int b)
    {
        return a - b;
    }
    public static void main(String args[])
    {
        System.out.println("My First Package");
    }
}
```

### 5. Write a java program to create a subpackage.

```
package p1.subp1;
public class program5
```

```

{
    public static int multiply(int a, int b)
    {
        return a * b;
    }
    public static void main(String args[])
    {
        System.out.println("My First Sub Package");
    }
}

```

**6. Write a java program to access the methods from package.**

```

import p1.program4;
public class program6
{
    public static void main(String args[])
    {
        System.out.println(program4.sum(5,6));
    }
}

```

**7. Write a java program to access the methods from package & subpackage both.**

```

import p1.program4;
import p1.subp1.program5;
public class program7
{
    public static void main(String args[])
    {
        System.out.println(program4.sum(5,6));
        System.out.println(program5.multiply(5,6));
    }
}

```

**8. Write a java program to observe the utility of public access specifier.**

```

import p1.program4;
public class program8
{
    public static void main(String args[])
    {
        System.out.println(program4.sum(5,6));
        System.out.println(program4.diff(5,6));
    }
}

```



## Assignment 8

### 1. Exception CASE-1: Problem without exception handling

```
public class program1
{
    static void calculate()
    {
        int a=1/0;
        System.out.println(a);
    }
    public static void main(String args[])
    {
        calculate();
        System.out.println("Thank You");
    }
}
```

### 2. Exception CASE-2: Exception creates but properly handled

```
public class program2
{
    static void calculate()
    {
        int a=1/0;
        System.out.println(a);
    }
    public static void main(String args[])
    {
        try
        {
            calculate();
        }
        catch(ArithmeticException e)
        {
            System.out.println("Denominator is 0");
        }
        System.out.println("Thank You");
    }
}
```

### 3. Exception CASE-3: Exception creates but not handled

```
public class program3
{
    static void calculate()
    {
        int a=1/0;
        System.out.println(a);
    }
    public static void main(String args[])
    {
        try
        {
            calculate();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Index not valid");
        }
        System.out.println("Thank You");
    }
}
```

#### 4. Write a java program to implement Java Multi catch block

```
public class program4
{
    static void calculate()
    {
        int a=1/0;
        System.out.println(a);
    }
    public static void main(String args[])
    {
        try
        {
            calculate();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Index not valid");
        }
        catch(ArithmeticException e)
        {
            System.out.println("Denominator is 0");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println("Thank You");
    }
}
```

#### 5. Write a java program to implement Java Nested try statement

```
public class program5
{
    static void calculate()
    {
        int a=1/0;
        System.out.println(a);
    }
    public static void main(String args[])
    {
        try
        {
            try
            {
                calculate();
            }
            catch(ArithmeticException e)
            {
                System.out.println("Denominator is 0");
            }
            try
            {
                int a=Integer.parseInt("S");
            }
            catch(ArrayIndexOutOfBoundsException e)
            {
                System.out.println("Index not valid");
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println("Thank You");
    }
}
```

```
    }  
}
```

**6. Write a java program to implement the throw keyword.**

```
public class program6  
{  
    static void calculate()  
    {  
        throw new ArithmeticException();  
    }  
    public static void main(String args[])  
    {  
        try  
        {  
            calculate();  
        }  
        catch(Exception e)  
        {  
            System.out.println(e);  
        }  
        System.out.println("Thank You");  
    }  
}
```

**7. Write a java program to implement the throws keyword.**

```
public class program7  
{  
    static void calculate() throws ArithmeticException  
    {  
        throw new ArithmeticException();  
    }  
    public static void main(String args[])  
    {  
        try  
        {  
            calculate();  
        }  
        catch(Exception e)  
        {  
            System.out.println(e);  
        }  
        System.out.println("Thank You");  
    }  
}
```

**8. Finally CASE – 1 : Case 1: finally example where exception doesn't occur.**

```
public class program8_1  
{  
    static void calculate()  
    {  
        System.out.println("Calculate called");  
    }  
    public static void main(String args[])  
    {  
        try  
        {  
            calculate();  
        }  
        catch(Exception e)  
        {  
            System.out.println(e);  
        }  
        finally
```

```

        {
            System.out.println("Finally Block");
        }
        System.out.println("Thank You");
    }
}

```

### **Case 2: finally example where exception occurs and not handled.**

```

public class program8_2
{
    static void calculate()
    {
        throw new ArithmeticException();
    }
    public static void main(String args[])
    {
        try
        {
            calculate();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Exception Occurred");
        }
        finally
        {
            System.out.println("Finally Block");
        }
        System.out.println("Thank You");
    }
}

```

### **Case 3: finally example where exception occurs and handled.**

```

public class program8_3
{
    static void calculate()
    {
        throw new ArithmeticException();
    }
    public static void main(String args[])
    {
        try
        {
            calculate();
        }
        catch(Exception e)
        {
            System.out.println("Exception Occurred");
        }
        finally
        {
            System.out.println("Finally Block");
        }
        System.out.println("Thank You");
    }
}

```

**9. Create a user-defined exception named CheckArgument to check the number of arguments passed through the command line. If the number of argument is less than 5, throw the CheckArgumentexception,else print the addition of all the five numbers.**

```
class CheckArguments extends Exception
{
    CheckArguments(String message)
    {super(message);}
}
class program9
{
    public static void main(String args[])
    {
        int sum=0;
        try
        {
            if(args.length!=5)
                throw new CheckArguments("Invalid number of inputs");
            try
            {
                for(int i=0;i<args.length;i++)
                    sum += Integer.parseInt(args[i]);
                System.out.println("Sum=" + sum);
            }
            catch(Exception e)
            {
                throw new CheckArguments("Invalid type of input");
            }
        }
        catch (CheckArguments e)
        {
            System.out.println(e);
        }
    }
}
```

**10. Consider a Student examination database system that prints the mark sheet of students. Input the following from the command line.**

**(a) Student's Name**

**(b) Marks in six subjects**

**These marks should be between 0 to 50. If the marks are not in the specified range, raise a RangeException, else find the total marks and prints the percentage of the students.**

```
import org.w3c.dom.ranges.RangeException;

class program10
{
    public static void main(String args[])
    {
        int sum=0;
        try
        {
            if(args.length!=7)
                throw new CheckArguments("Invalid number of inputs");
            try
            {
                for (int i = 1; i < 7; i++) {
                    if (Integer.parseInt(args[i]) >= 0 && Integer.parseInt(args[i]) <= 50)
                        sum += Integer.parseInt(args[i]);
                    else
                        throw new RangeException((short) 0, "Range of marks must be between 0 to 50");
                }
            }
        }
    }
}
```

```
        System.out.println("Total marks of "+args[0]+" is : " + sum);
        System.out.println("Total Percentage of "+args[0]+" is : " + sum/6.0*2);
    }
    catch(RangeException e)
    {
        System.out.println(e);
    }
        catch(Exception e)
        {
            throw new CheckArguments("Invalid type of input");
        }
    }
    catch (CheckArguments e)
    {
        System.out.println(e);
    }
}
}
```

## Assignment 9

### 1. Write a java program to create Thread by extending the Thread class.

```
class MyThread extends Thread {
    public void run() {
        System.out.println("MyThread is running.");
    }
}

public class program1 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();
    }
}
```

### 2. Write a java program to create Thread by implementing the Runnable interface.

```
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("MyRunnable is running.");
    }
}

public class program2 {
    public static void main(String[] args) {
        MyRunnable runnable = new MyRunnable();
        Thread thread = new Thread(runnable);
        thread.start();
    }
}
```

### 3. Write a java program to implement the yield( ) method in thread programming.

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("MyThread is running: " + i);
            Thread.yield();
        }
    }
}

public class program3 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();

        for (int i = 1; i <= 5; i++) {
            System.out.println("Main thread is running: " + i);
        }
    }
}
```

### 4. Write a java program to implement the sleep(n) method in thread programming.

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            try {
                Thread.sleep(1000); // sleep for 1 second
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

        }
        System.out.println("MyThread is running: " + i);
    }
}

}

public class program4 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();

        for (int i = 1; i <= 5; i++) {
            System.out.println("Main thread is running: " + i);
        }
    }
}

```

**5. Write a java program to implement the suspend( ) & resume( ) method in thread programming.**

```

class MyThread extends Thread {
    private boolean suspended = false;

    public void run() {
        for (int i = 1; i <= 5; i++) {
            try {
                Thread.sleep(1000); // sleep for 1 second
                synchronized(this) {
                    while (suspended) {
                        wait();
                    }
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("MyThread is running: " + i);
        }
    }

    public void suspendThread() {
        suspended = true;
    }

    public synchronized void resumeThread() {
        suspended = false;
        notify();
    }
}

public class program5 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();

        try {
            Thread.sleep(3000); // sleep for 3 seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        thread.suspendThread();

        try {
            Thread.sleep(3000); // sleep for 3 seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```



```

    }

    thread.resumeThread();
}
}

```

**6. Write a java program to implement the sleep(n) method in thread programming.**

```

class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            try {
                Thread.sleep(1000); // sleep for 1 second
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("MyThread is running: " + i);
        }
    }
}

public class program6 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();

        for (int i = 1; i <= 5; i++) {
            System.out.println("Main thread is running: " + i);
        }
    }
}

```

**7. Create 4 threads with priority 1,3,5,7 respectively. Update a counter in each of the threads for 10 ms. Print the final value of count for each thread.**

```

class CounterThread extends Thread {
    private int count;

    public CounterThread(int priority) {
        setPriority(priority);
    }

    public void run() {
        for (int i = 0; i < 100; i++) {
            count++;
            try {
                Thread.sleep(10); // sleep for 10 milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Thread " + getPriority() + " count: " + count);
    }
}

public class program7 {
    public static void main(String[] args) {
        CounterThread thread1 = new CounterThread(1);
        CounterThread thread2 = new CounterThread(3);
        CounterThread thread3 = new CounterThread(5);
        CounterThread thread4 = new CounterThread(7);

        thread1.start();
        thread2.start();
        thread3.start();
    }
}

```

```
        thread4.start();
    }
}
```

**8. Write a Java Program to Synchronize the Threads Acting on the Same Object. The Synchronized Block in the Program can be Executed by Only One Thread at a Time.**

```
class Counter {
    private int count;

    public synchronized void increment() {
        count++;
    }

    public synchronized int getCount() {
        return count;
    }
}

class CounterThread extends Thread {
    private Counter counter;

    public CounterThread(Counter counter) {
        this.counter = counter;
    }

    public void run() {
        for (int i = 0; i < 10000; i++) {
            counter.increment();
        }
    }
}

public class program8 {
    public static void main(String[] args) {
        Counter counter = new Counter();
        CounterThread thread1 = new CounterThread(counter);
        CounterThread thread2 = new CounterThread(counter);

        thread1.start();
        thread2.start();

        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Final count: " + counter.getCount());
    }
}
```

**9. Write a Java Program to Check a Thread is Alive or Not.**

```
class MyThread extends Thread {
    public void run() {
        try {
            Thread.sleep(500); // sleep for 5 seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

```

public class program9 {
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        System.out.println("Thread is alive: " + thread.isAlive());
        thread.start();
        System.out.println("Thread is alive: " + thread.isAlive());
        try {
            Thread.sleep(1000); // sleep for 1 second
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("Thread is alive: " + thread.isAlive());
    }
}

```

#### 10. Write a Java Program to Get the Name of a Running Thread.

```

class MyThread extends Thread {
    public void run() {
        System.out.println("Thread name: " + Thread.currentThread().getName());
    }
}

public class program10 {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();
        thread1.start();
        thread2.start();
    }
}

```

## Assignment 10

### 1. Write a java program to implement the concept of Applet.

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

public class program1 extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello, world!", 50, 50);
    }
}
```

### 2. Write a java program to implement the parameter passing applet.

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

public class program2 extends Applet {
    private String message;

    public void init() {
        message = getParameter("message");
    }

    public void paint(Graphics g) {
        g.drawString(message, 20, 20);
    }
}
```

### 3. Write a java program to implement JButton Class.

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class program3 implements ActionListener {
    private JButton myButton;

    public program3() {
        myButton = new JButton("Click me!");
        myButton.addActionListener(this);
        JPanel myPanel = new JPanel();
        myPanel.add(myButton);
        JFrame myFrame = new JFrame("My JButton Example");
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.add(myPanel);
        myFrame.pack();
        myFrame.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == myButton) {
            System.out.println("Button clicked!");
        }
    }

    public static void main(String[] args) {
        new program3();
    }
}
```

#### 4. Write a java program to implement JTextField Class.

```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class program4 implements ActionListener {
    private JTextField myTextField;

    public program4() {
        myTextField = new JTextField(10);
        myTextField.addActionListener(this);
        JPanel myPanel = new JPanel();
        myPanel.add(new JLabel("Enter text: "));
        myPanel.add(myTextField);
        JFrame myFrame = new JFrame("My JTextField Example");
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.add(myPanel, BorderLayout.CENTER);
        myFrame.pack();
        myFrame.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == myTextField) {
            System.out.println("Text entered: " + myTextField.getText());
            myTextField.setText("");
        }
    }

    public static void main(String[] args) {
        new program4();
    }
}
```

#### 5. Write a java program to implement JPanel Class.

```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import java.awt.BorderLayout;

public class program5 {
    public program5() {
        JPanel myPanel = new JPanel();
        myPanel.add(new JLabel("Hello, world!"));
        JFrame myFrame = new JFrame("My JPanel Example");
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.add(myPanel, BorderLayout.CENTER);
        myFrame.pack();
        myFrame.setVisible(true);
    }

    public static void main(String[] args) {
        new program5();
    }
}
```

#### 6. Write a java program to implement JMenu Class.

```
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
```

```

import javax.swing.JMenuItem;

public class program6 {
    public program6() {
        JFrame myFrame = new JFrame("My JMenu Example");
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JMenuBar myMenuBar = new JMenuBar();
        JMenu fileMenu = new JMenu("File");
        JMenuItem newOption = new JMenuItem("New");
        JMenuItem openOption = new JMenuItem("Open");
        fileMenu.add(newOption);
        fileMenu.add(openOption);
        myMenuBar.add(fileMenu);
        myFrame.setJMenuBar(myMenuBar);
        myFrame.pack();
        myFrame.setVisible(true);
    }

    public static void main(String[] args) {
        new program6();
    }
}

```

**7. Write a java program to implement Chat Frame comprising JFrame, JMenuBar, JMenu, JMenuItem, JPanel, JLabel, JTextField, JButton etc.**

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class program7 extends JFrame implements ActionListener {
    private JTextArea chatArea;
    private JTextField messageField;
    private JButton sendButton;
    public program7() {
        setTitle("Chat Frame");
        JMenuBar menuBar = new JMenuBar();
        JMenu fileMenu = new JMenu("File");
        JMenuItem exitMenuItem = new JMenuItem("Exit");
        fileMenu.add(exitMenuItem);
        menuBar.add(fileMenu);
        setJMenuBar(menuBar);
        JPanel chatPanel = new JPanel(new BorderLayout());
        chatArea = new JTextArea();
        chatArea.setEditable(false);
        chatPanel.add(chatArea, BorderLayout.CENTER);
        JPanel messagePanel = new JPanel(new BorderLayout());
        messageField = new JTextField();
        messagePanel.add(messageField, BorderLayout.CENTER);
        sendButton = new JButton("Send");
        sendButton.addActionListener(this);
        messagePanel.add(sendButton, BorderLayout.EAST);
        chatPanel.add(messagePanel, BorderLayout.SOUTH);
        add(chatPanel);
        setSize(400, 400);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        String message = messageField.getText();
        chatArea.append("You: " + message + "\n");
        messageField.setText("");
    }
    public static void main(String[] args) {
        new program7();
    }
}

```