1.1) Write a Java Program to display Default values of all primitive data types of Java.

Aim: Java Program to print default values of all primitive data types in Java.

Program:

```
class DefaultVals
{
        static byte b;
        static short s;
        static int i;
        static long l;
        static float f;
        static double d;
        static char c;
        static boolean bl;
        public static void main(String[] args)
        {
                System.out.println("The default values of primitive data types are:");
                System.out.println("Byte :"+b);
                System.out.println("Short :"+s);
                System.out.println("Int :"+i);
                System.out.println("Long :"+l);
                System.out.println("Float :"+f);
                System.out.println("Double :"+d);
                System.out.println("Char :"+c);
                System.out.println("Boolean :"+bl);
        }
}
```

Output:

```
D:\21A95A0503>javac DefaultVals.java

D:\21A95A0503>java DefaultVals
The default values of primitive data types are:
Byte :0
Short :0
Int :0
Long :0
Float :0.0
Double :0.0
Char :
Boolean :false
```

1.2) Write a Java program to find the discriminant value D and find out the roots of the quadratic equation of the form ax2+bx+c=0

Aim: Java Program to find discriminant value D and find roots of quadratic equations of form ax2+bx+c=0

Program:

```
public class QuadEq
 {
   public static void main(String[] Strings)
   {
      double a=Double.parseDouble(Strings[0]);
      double b=Double.parseDouble(Strings[1]);
      double c=Double.parseDouble(Strings[2]);
      double d=b *b -4.0 * a *c;
      System.out.println("Discriminent Value:" +Math.pow(d,0.5));
      if(Double.isNaN(d))
         System.out.println("Equation has no roots");
      if(d>0.0)
      {
         double r1=(-b+Math.pow(d,0.5))/(2.0*a);
         double r2=(-b-Math.pow(d,0.5))/(2.0*a);
         System.out.println("The roots are"+r1+"and"+r2);
      }
      else if(d==0.0)
      {
         double r1=-b/(2.0*a);
         System.out.println("The root is:"+r1);
      }
      else
      {
         System.out.println("Roots are not real");
      }
```

```
    }
  }
```

Output:

```
D:\21A95A0503>javac QuadEq.java

D:\21A95A0503>java QuadEq 1 5 4
Discriminent Value:3.0
The roots are-1.0and-4.0
```

1.3) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Aim: Java Program to print the racer whose speed is greater than the average speed.

Program:

```
public class Bikers
 {
   public static void main (String[] Strings)
   {
      double a=Double.parseDouble(Strings[0]);
      double b=Double.parseDouble(Strings[1]);
      double c=Double.parseDouble(Strings[2]);
      double d=Double.parseDouble(Strings[3]);
      double e=Double.parseDouble(Strings[4]);
      double avg_speed=(a+b+c+d+e)/5;
      System.out.println("the average speed for qualifying racers are:"+avg_speed);
      if(a>avg_speed)
         System.out.println("racer 1 is qualified with speed "+a+"which is more than avg
speed"+avg_speed);
      if(b>avg_speed)
         System.out.println("racer 2 is qualified with speed "+b+"which is more than avg
speed"+avg_speed);
      if(c>avg_speed)
         System.out.println("racer 3 is qualified with speed "+c+"which is more than avg
speed"+avg_speed);
      if(d>avg_speed)
         System.out.println("racer 4 is qualified with speed "+d+"which is more than avg
speed"+avg_speed);
      if(e>avg_speed)
         System.out.println("racer 5 is qualified with speed "+e+"which is more than avg
speed"+avg_speed);
```

```
    else
        System.out.println("none of the racers qualify the race:");


  }
}
```

Output:

```
D:\21A95A0503>javac Bikers.java

D:\21A95A0503>java Bikers 10 20 30 40 50
the average speed for qualifying racers are:30.0
racer 4 is qualified with speed 40.0which is more than avg speed30.0
racer 5 is qualified with speed 50.0which is more than avg speed30.0
```

2.1) Write a Java program to select all the prime numbers within the range of 1to100.

Aim: Write a Java program to select all the prime numbers within the range of 1to100.

Program:

```
public class PrimeNums
 {
   public static void main(String[] args)
   {
      int ct=0,n=0,i=1,j=1;
      while(n<25){
         j=1;
         ct=0;
         while(j<=i){
            if(i%j==0)
               ct++;
            j++;
         }
         if(ct==2){
            System.out.printf("%d ",i);
            n++;
         }
         i++;
      }
   }
}
```

Output:

```
D:\21A95A0503>javac PrimeNums.java

D:\21A95A0503>java PrimeNums

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

ADITYA
ENLIGHTENS THE NESCIENCE

2.2) Write a Java program to Find the sum of all even terms in the Fibonacci sequence up to the given range N.

Aim: Write a Java program to Find the sum of all even terms in the Fibonacci sequence up to the given range N.

Program:

```
import java.util.Scanner;

public class Fibonacci

{

   public static void main(String args[])

   {

      int f=0,s=1;

      int t=0,sum=0;

      int a[]=new int[100];

      Scanner input=new Scanner(System.in);

      a[0]=0;

      a[1]=1;

      System.out.println("enter rhe range for fibonacci series:");

      int range=input.nextInt();

      System.out.println("the fibonacci series is as follows:");

      System.out.print(0);

      System.out.print(" "+1);

      for(int i=2;i<range;i++)

      {

         t=f+s;

         System.out.print(" "+t);

         a[i]=t;

         f=s;

           s=t;

       }

      for(int i=2;i<=range;i++)
```

```
    {
        if(i%2==0)
            sum=sum+a[i];
    }
    System.out.println("\n"+"Sum of Even Terms in Fibonacci Series upto a range "+range+" is:"+sum);
  }
 }
```

Output:

```
D:\21A95A0503>javac Fibonacci.java

D:\21A95A0503>java Fibonacci
Enter the range for fibonacci series:
11
The fibonacci series is as follows:
0 1 1 2 3 5 8 13 21 34 55
Sum of Even Terms in Fibonacci Series upto a range 11 is:88
```

2.3) Write a Java program to check whether a given number is Armstrong or not.

Aim: To write a Java program to check whether a given number is Armstrong or not.

Program:

```
public class Armstrong
{
   public static void main (String[] args)
   {
      int number=1634,originalNumber,remainder,result=0,n=0;
      originalNumber=number;
      for(;originalNumber!=0;originalNumber/=10,++n);
      originalNumber=number;
      for(;originalNumber!=0;originalNumber/=10)
      {
         remainder=originalNumber%10;
         result+=Math.pow(remainder, n);
      }
      if(result==number)
         System.out.println(number +" is an armstrong number");
      else
         System.out.println(number +" is not an armstrong number");
   }
 }
```

Output:

```
D:\21A95A0503>javac Armstrong.java

D:\21A95A0503>java Armstrong
1634 is an armstrong number
```

3.1) Write a Java program to implement binary search.

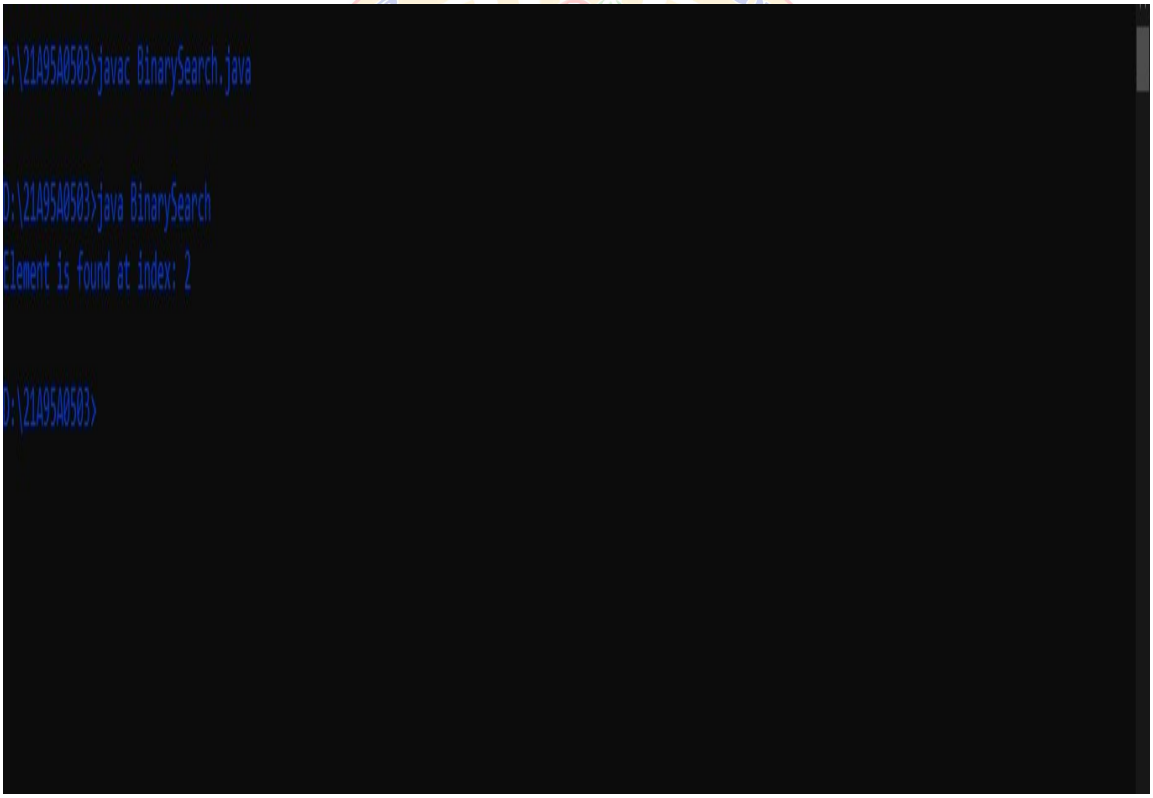
Aim: Write a Java program to implement binary search.

Program:

```
class BinarySearch
{
   public static int binarySearch(int arr[], int first, int last, int key)
   {
         if (last>=first)
         {
         int mid = first + (last - first)/2;
         if (arr[mid] == key)
         {
         return mid;
         }
         if (arr[mid] > key)
         {
         return binarySearch(arr, first, mid-1, key);
         }
         else
         {
         return binarySearch(arr, mid+1, last, key);
         }
         }
      return -1;
}
   public static void main(String args[])
   {
```

```
        int arr[] = {10,20,30,40,50};
        int key = 30;
        int last=arr.length-1;
        int result = binarySearch(arr,0,last,key);
        if (result == -1)
            System.out.println("Element is not found!");
        else
            System.out.println("Element is found at index: "+result);
    }
}
```

Output:



3.2) Write a Java program to sort for an element in a given list of elements using bubble sort.

Aim: Write a Java program to sort for an element in a given list of elements using bubble sort.

Program:

```
class BubbleSort
{
        public static void main(String args[])
        {
                int a[]={31,19,9,23,41};
                int i,j,temp;
                for(i=0;i<5-1;i++)
                {
                        for(j=0;j<5-1-i;j++)
                        {
                                if(a[j]>a[j+1])
                                {
                                        temp=a[j];
                                        a[j]=a[j+1];
                                        a[j+1]=temp;
                                }
                        }
                }
                System.out.println("Sorted elements are:");
                for(i=0;i<a.length;i++)
                {
                        System.out.print(a[i]+" ");
                }
```

```
        }
}
```

Output:


D:\21A95A0503>javac BubbleSort.java
D:\21A95A0503>java BubbleSort
Sorted elements are:
9 19 23 31 41
D:\21A95A0503>

3.3) Write a Java program to sort for an element in a given list of elements using merge sort.

Aim: Write a Java program to sort for an element in a given list of elements using merge sort.

Program:

```
class MergeSort
{
void merge(int a[], int beg, int mid, int end)
{
   int i, j, k;
   int n1 = mid - beg + 1;
   int n2 = end - mid;
   int LeftArray[] = new int[n1];
   int RightArray[] = new int[n2];
   for (i = 0; i < n1; i++)
   LeftArray[i] = a[beg + i];
   for (j = 0; j < n2; j++)
   RightArray[j] = a[mid + 1 + j];
   i = 0;
   j = 0;
   k = beg;

   while (i < n1 && j < n2)
   {
      if(LeftArray[i] <= RightArray[j])
      {
         a[k] = LeftArray[i];
         i++;
```

```
        }
        else
        {
            a[k] = RightArray[j];
            j++;
        }
        k++;
    }
    while (i<n1)
    {
        a[k] = LeftArray[i];
        i++;
        k++;
    }

    while (j<n2)
    {
        a[k] = RightArray[j];
        j++;
        k++;
    }
}

void mergeSort(int a[], int beg, int end)
{
    if (beg < end)
    {
```

```
        int mid = (beg + end) / 2;
        mergeSort(a, beg, mid);
        mergeSort(a, mid + 1, end);
        merge(a, beg, mid, end);
    }
}

void printArray(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        System.out.print(a[i] + " ");
}

public static void main(String args[])
{
    int a[] = { 11, 30, 24, 7, 31, 16, 39, 41 };
    int n = a.length;
    MergeSort m1 = new MergeSort();
    System.out.println("\nBefore sorting array elements are - ");
    m1.printArray(a, n);
    m1.mergeSort(a, 0, n - 1);
    System.out.println("\nAfter sorting array elements are - ");
    m1.printArray(a, n);
    System.out.println("");
}
}
```

Output:

```
D:\21A95A0503>javac MergeSort.java

D:\21A95A0503>java MergeSort

Before sorting array elements are -
11 30 24 7 31 16 39 41
After sorting array elements are -
7 11 16 24 30 31 39 41

D:\21A95A0503>
```

4.1) Write a Java program to display the details of a person. Personal details should be given in one method and the qualification details in another method.

Aim: Write a Java program to display the details of a person. Personal details should be given in one method and the qualification details in another method.

Program:

```
class Details
{
        void details()
        {
                String name="Nikhil";
                int age=19;
                String gender="Male";
                System.out.println("Name of the person:"+name);
                System.out.println("Age of the person:"+age);
                System.out.println("Gender of the person:"+gender);
        }
        void qualification()
        {
                String quali="Diploma";
                System.out.println("Qualification of the person:"+quali);
        }

}
class Show
{
        public static void main(String args[])
        {
                Details d=new Details();
```

```
            d.details();
            d.qualification();
        }
}
```

Output:

```
D:\21A95A0503>javac Show.java

D:\21A95A0503>java Show
Name of the person:Nikhil
Age of the person:19
Gender of the person:Male
Qualification of the person:Diploma
```

4.2) Write a Java program to implement constructor and constructor overloading

Aim: Write a Java program to implement constructor and constructor overloading

Program:

Constructor:

```
class Company
{
  String name;
   Company()
  {
    name = "Infosys";
  }
}

class Main
{
  public static void main(String[] args)
  {
    Company obj = new Company();
    System.out.println("Company name = " + obj.name);
  }
}
```

Output:

```
D:\21A95A0503>javac Main.java

D:\21A95A0503>java Main
Company name = Infosys

D:\21A95A0503>
```

Constructor Overloading:

```
class Box
{
        int width,height,depth;
        Box()
        {
                width=10;
                height=20;
                depth=30;
        }
        Box(int w,int h,int d)
        {
                width=w;
                height=h;
                depth=d;
        }
        Box(int x)
        {
                width=x;
                height=x;
                depth=x;
        }
        void volume()
        {
                System.out.println(width*height*depth);
        }
}
class DisCon
{
        public static void main(String args[])
```

```
        {
                Box b1=new Box();
                Box b2=new Box(2,5,19);
                Box b3=new Box(27);
                b1.volume();
                b2.volume();
                b3.volume();
        }
}
```

Output:

```
D:\21A95A0503>javac DisCon.java

D:\21A95A0503>java DisCon
6000
190
19683

D:\21A95A0503>
```

4.3) Write a Java program to implement method overloading.

Aim: Write a Java program to implement method overloading.

Program:

```
class MethodOverloading
{

   private static void display(int a)
   {
      System.out.println("Got Integer data.");
   }

   private static void display(String a)
   {
      System.out.println("Got String object.");
   }

   public static void main(String[] args)
   {
      display(1);
      display("Hello");
   }
}
```

Output:

```
D:\21A95A0503>javac MethodOverloading.java

D:\21A95A0503>java MethodOverloading
Got Integer data.
Got String object.

D:\21A95A0503>
```

ENLIGHTENS THE NESCIENCE