Part A

1. Write a Java program to find the Armstrong Number.
2. Write a java program that uses both recursive and non-recursive functions to print the Fibonacci Sequence.
3. Java program to remove duplicate elements in a sorted array.
4. Java Program to count the total number of vowels and consonants in a string.
5. Program to create a class representing a Circle with attributes radius and methods to calculate area and circumference. Create an object and display the results.
6. Program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum() to add two distances by using objects as method arguments and show the result. (Input and output of inches should be less than 12.)
7. Write a Program to illustrate Inheritance in Java.
8. Write a Java program to demonstrate constructor overloading.

Part A

1. Write a Java program to find the Armstrong Number.

```
import java.util.*;
public class ArmNum
{
	//function to check if the number is Armstrong or not
static boolean isArmstrong(int n)
{
int temp, remainder=0, sum=0;
	//assigning n into a temp variable
temp=n;
	//to get the number of digits in the number entered by the user
int digits = String.valueOf(n).length();
while(temp>0)
{
	//determines the last digit from the number
remainder = temp % 10;
	//calculates the power of a number up to digit times and add the resultant to the
	sum variable
sum +=  (Math.pow(remainder, digits));
	//removes the last digit
temp = temp/10;
}
	//compares the sum with n
if(n==sum)
	//returns if sum and n are equal
return true;
	//returns false if sum and n are not equal
else return false;
}

public static void  main(String args[])
{
int num;
Scanner sc= new Scanner(System.in);
System.out.print("Enter the number: ");
	//reads the limit from the user
num=sc.nextInt();

if(isArmstrong(num))
{
System.out.print("The number is Armstrong");
```

```
}
else
{
System.out.print("Not Armstrong ");
}
}
}
```

```
C:\Users\HP\Desktop\YenJava>java ArmNum
Enter the number: 567
Not Armstrong
C:\Users\HP\Desktop\YenJava>java ArmNum
Enter the number: 153
The number is Armstrong
C:\Users\HP\Desktop\YenJava>
```

Explanation:

Armstrong Number:

An **Armstrong** number is a positive m-digit number that is equal to the sum of the m^{th} powers of their digits. Let's understand it through an example.

Armstrong Number Example

1: 1^1 = **1**

2: 2^1 = **2**

3: 3^1 = **3**

153: $1^3 + 5^3 + 3^3 = 1 + 125 + 27$ = **153**

125: $1^3 + 2^3 + 5^3 = 1 + 8 + 125$ = **134 (Not an Armstrong Number)**

1634: $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256$ = **1643**

The first few Armstrong numbers between 0 to 999 are **1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407.**

2. Write a java program that uses both recursive and non-recursive functions to print the Fibonacci Sequence.

```
import java.util.Scanner;
class Series
{
 int F1, F2=1,F3=0;
short count;
void nonrecursive(short n)
{
count=0;
F1=0;
F2=1;
F3=0;
while (count<n)
{
System.out.println(F1);
F3=F1+F2;
F1=F2;
F2=F3;
count++;
}
}
void recursive(short n)
{
 int i=0;
for ( int c = 1 ; c <= n ; c++ )
{
System.out.println(Fib(i));
i++;
}
}
int Fib(int n)
{
if ( n == 0 )
return 0;
else if ( n == 1 )
return 1;
else
return ( Fib(n-1) + Fib(n-2) );
}
}
```

```
class Fibonacci
{
public static void main(String args[])
{
System.out.println("Enter the number n to print the fabonicci series : ");
Scanner sc=new Scanner(System.in);
short n=sc.nextShort();
Series ob=new Series();
System.out.println("First " + n + " Fibonacci numbers using recursive function");
ob.recursive(n);
System.out.println("First " + n + " Fibonacci numbers using non-recursive function");
ob.nonrecursive(n);
}
}
```

```
C:\Users\HP\Desktop\YenJava>java Fibonacci
Enter the number n to print the fabonicci series :
6
First 6 Fibonacci numbers using recursive function
0
1
1
2
3
5
First 6 Fibonacci numbers using non-recursive function
0
1
1
2
3
5
```

3. Java program to remove duplicate elements in a sorted array.

```
public class RemvDuplicate
 {

                                // Method to remove duplicate elements from a sorted array
   public static int removeDuplicates(int[] arr)
   {
      if (arr.length == 0 || arr.length == 1)
     {
         return arr.length;              // Return length as it is if array has 0 or 1 element
      }

      int j = 0;                          // Pointer for the position of unique elements
      for (int i = 0; i < arr.length - 1; i++)
       {
         if (arr[i] != arr[i + 1])
        {                                // If current element is different from the next one
            arr[j++] = arr[i];           // Store the unique element
         }
      }

      arr[j++] = arr[arr.length - 1];  // Add the last elmnt as it is always unique in a sorted array
      return j;                          // Return the count of unique elements
    }

   public static void main(String[] args)
   {
      int[] arr = {10, 20, 20, 30, 30, 40, 50, 50};
      int newLength = removeDuplicates(arr);

      // Print the unique elements
      System.out.println("Array after removing duplicates:");
      for (int i = 0; i < newLength; i++)
     {
         System.out.print(arr[i] + " ");
      }
   }
}
```

```
C:\Users\HP\Desktop\YenJava>javac RemvDuplicate.java

C:\Users\HP\Desktop\YenJava>java RemvDuplicate
Array after removing duplicates:
10 20 30 40 50
```

4. Java Program to count the total number of vowels and consonants in a string.

```
public class CountVowelConsonant
{
   public static void main(String[] args)
  {
                              //Counter variable to store the count of vowels and consonant
      int vCount = 0, cCount = 0;

                              //Declare a string
      String str = "This is a really simple sentence";

                              //Converting entire string to lower case to reduce the comparisons
      str = str.toLowerCase();

      for(int i = 0; i < str.length(); i++)
     {
                               //Checks whether a character is a vowel
         if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' ||
str.charAt(i) == 'u')
        {
                              //Increments the vowel counter
           vCount++;
         }
                              //Checks whether a character is a consonant
         else if(str.charAt(i) >= 'a' && str.charAt(i)<='z')
         {
                               //Increments the consonant counter
           cCount++;
         }
      }
      System.out.println("Number of vowels: " + vCount);
      System.out.println("Number of consonants: " + cCount);
   }
}
```

```
C:\Users\HP\Desktop\YenJava>javac CountVowelConsonant.java

C:\Users\HP\Desktop\YenJava>java CountVowelConsonant
Number of vowels: 10
Number of consonants: 17
```

5. Write a Java program to create a class representing a Circle with attributes radius and methods to calculate area and circumference. Create an object and display the results. [class basics]

```
class Circle {
   private double radius;

                    // Constructor to initialize radius
   public Circle(double radius) {
      this.radius = radius;
   }

                    // Method to calculate area
   public double calculateArea() {
      return Math.PI * radius * radius;
   }

                    // Method to calculate circumference
   public double calculateCircumference() {
      return 2 * Math.PI * radius;
   }
                    // Getter for radius
   public double getRadius() {
      return radius;
   }
                    // Setter for radius
   public void setRadius(double radius) {
      this.radius = radius;
   }
}

public class CircleDemo {
   public static void main(String[] args) {
                     // Create a Circle object with radius 5.0
      Circle circle = new Circle(5.0);

                    // Calculate area and circumference
      double area = circle.calculateArea();
      double circumference = circle.calculateCircumference();

                    // Display results
      System.out.println("Circle with radius: " + circle.getRadius());
      System.out.println("Area: " + area);
```

```
        System.out.println("Circumference: " + circumference);
    }
}
```

```
C:\Users\HP\Desktop\YenJava>javac CircleDemo.java

C:\Users\HP\Desktop\YenJava>java CircleDemo
Circle with radius: 5.0
Area: 78.53981633974483
Circumference: 31.41592653589793
```

6. Program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum ( ) to add two distances by using objects as method arguments and show the result. (Input and output of inches should be less than 12.) [constructors]

```
import java.util.Scanner;
class distance
{
    int feet;
    int inches;
    distance()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter feet:");
        feet=sc.nextInt();
        System.out.println("Enter inches:");
        inches=sc.nextInt();
    }
    public void showDistance()
    {
        System.out.println("Feet:"+feet+"\tInches:"+inches);
    }
    void sum(distance D1,distance D2)
    {
        inches=D1.inches+D2.inches;
        feet=D1.feet+D2.feet+(inches/12);
        inches=inches%12;
    }
}

public class labA5
{
    public static void main(String[] s)
    {
```

```
      distance D1=new distance();
      distance D2=new distance();
      System.out.println("second distance:");
      D1.showDistance();
      System.out.println("second distance:");
      D2.showDistance();
      D1.sum(D1,D2);
      System.out.println("Total distance is:");
      D1.showDistance();
   }
}
```

```
C:\Users\HP\Desktop\YenJava>javac DistancePrg.java

C:\Users\HP\Desktop\YenJava>java DistancePrg
Enter feet:
7
Enter inches:
9
Enter feet:
8
Enter inches:
8
second distance:
Feet:7  Inches:9
second distance:
Feet:8  Inches:8
Total distance is:
Feet:16 Inches:5
```

7. Write a Program to illustrate Inheritance in Java.

```
class TwoDshape
{
private double width;
private double height;
   //Accessor methods for width and height
double getWidth( )
{
   return width;
}
double getHeight( )
{
   return height;
}
void setWidth(double w )
{
```

```
   width=w;
}
void setHeight(double h )
{
   height=h;
}
void showDim ( )
{
System.out.println( "Width and height are " + width + "and"  +height);
}
}
          //A subclass of TwoDshape for triangle.
class Triangle extends TwoDshape
{
private String style;
          //Constructor
Triangle (String s, double w, double h)
{
setWidth(w);
setHeight(h);
style=s;
}
double area ( )
{
return getWidth ( ) * getHeight ( ) / 2;
}
void showStyle ( )
{
   System.out.println("Triangle is "+ style);
}
}
class ShapesExample
{
public static void main (String args [ ])
{
Triangle t1 = new Triangle ( "filled", 4.0, 4.0);
Triangle t2 = new Triangle ( "outlined", 8.0, 12.0);
System.out.println("Info for t1: ");
t1.showStyle ( );
t1.showDim ( );
System.out.println("Area is " + t1.area ( ));
System.out.println ( );
System.out.println("Info for t2: ");
t2.showStyle ( );
```

```
t2.showDim ( );
System.out.println("Area is " + t2.area ( ));
}
}
```

```
C:\Users\HP\Desktop\YenJava>javac ShapesExample.java

C:\Users\HP\Desktop\YenJava>java ShapesExample
Info for t1:
Triangle is filled
Width and height are 4.0and4.0
Area is 8.0

Info for t2:
Triangle is outlined
Width and height are 8.0and12.0
Area is 48.0
```

8. Write a Java program to demonstrate constructor overloading. [polymorphism]

```
class Rectangle
{
int length;
int width;
Rectangle ( )
{
length= 10;
width = 20;
}
Rectangle ( int k)
{
length = k; width = k;
}
Rectangle (int x, int y) //Constructor method
{
        length = x ;
        width = y;
}
int rectArea ( )
{
return(length * width);
}
}

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

```
{
Rectangle rect1= new Rectangle ( );
Rectangle rect2 = new Rectangle (10);
Rectangle rect3 = new Rectangle (15,10);
int areal = rect1.rectArea( ) ;
System.out.println("Area1 = "+ areal) ;
int area2 = rect2.rectArea( ) ;
System.out.println("Area2 = "+ area2) ;
int area3 = rect3.rectArea( ) ;
System.out.println("Area3 = "+ area3) ;
}
}
```

```
C:\Users\HP\Desktop\YenJava> javac RectangleArea.java

C:\Users\HP\Desktop\YenJava> java RectangleArea
Area1 = 200
Area2 = 100
Area3 = 150
```