

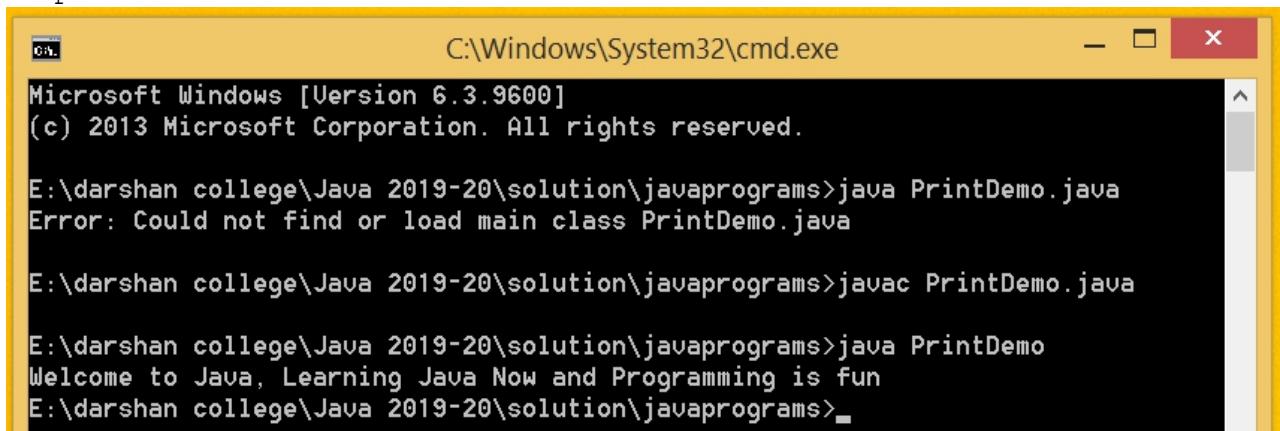
3140705 – Object Oriented Programming -I

Practical-1

2. WAP to print “Welcome to Java, Learning Java Now and Programming is fun”.

```
class PrintDemo{  
    //Main Method  
    public static void main(String[] args) {  
  
        //print string using print method  
        System.out.print("Welcome to Java, Learning Java Now and  
        Programming is fun");  
  
    }  
}
```

Output:



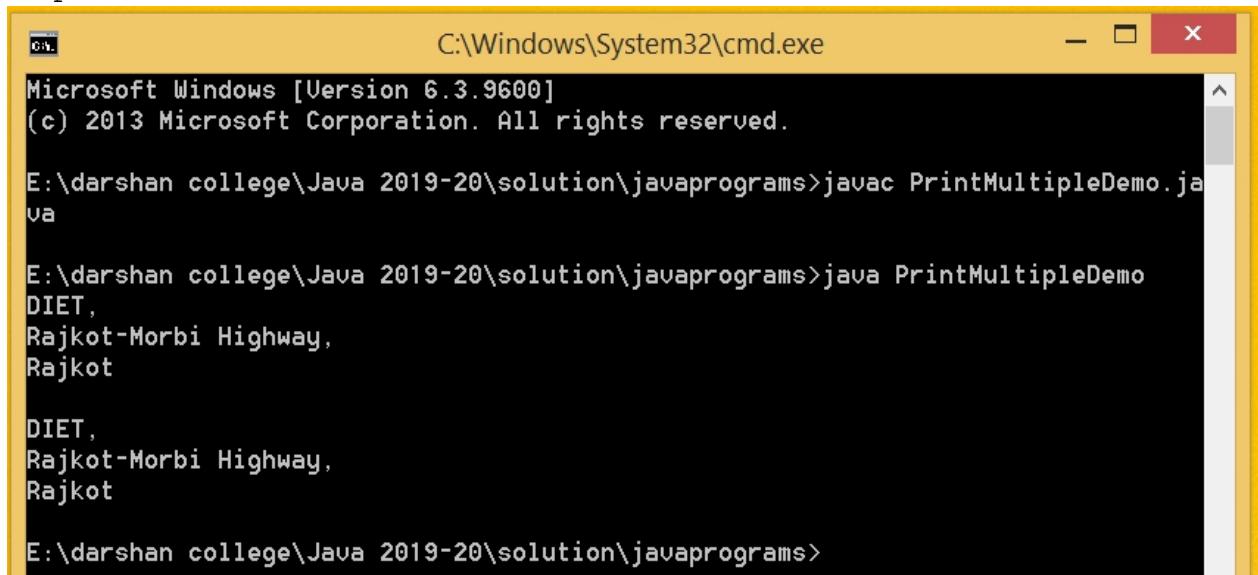
```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
  
E:\darshan college\Java 2019-20\solution\javaprograms>java PrintDemo.java  
Error: Could not find or load main class PrintDemo.java  
  
E:\darshan college\Java 2019-20\solution\javaprograms>javac PrintDemo.java  
  
E:\darshan college\Java 2019-20\solution\javaprograms>java PrintDemo  
Welcome to Java, Learning Java Now and Programming is fun  
E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. WAP to print your address i) using single print ii) using multiple println

```
class PrintMultipleDemo{  
    //Main Method  
    public static void main(String[] args) {  
  
        //print string using single print  
        System.out.print("DIET,\nRajkot-Morb Highway,\nRajkot\n");  
  
        //print string using multiple println  
        System.out.println("DIET,");  
        System.out.println("Rajkot-Morbi Highway,");  
        System.out.println("Rajkot");  
    }  
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac PrintMultipleDemo.ja
va

E:\darshan college\Java 2019-20\solution\javaprograms>java PrintMultipleDemo
DIET,
Rajkot-Morbi Highway,
Rajkot

DIET,
Rajkot-Morbi Highway,
Rajkot

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-2

1. WAP to print addition of 2 numbers (with Scanner)

```
import java.util.Scanner;
public class AdditionTwoNumber {

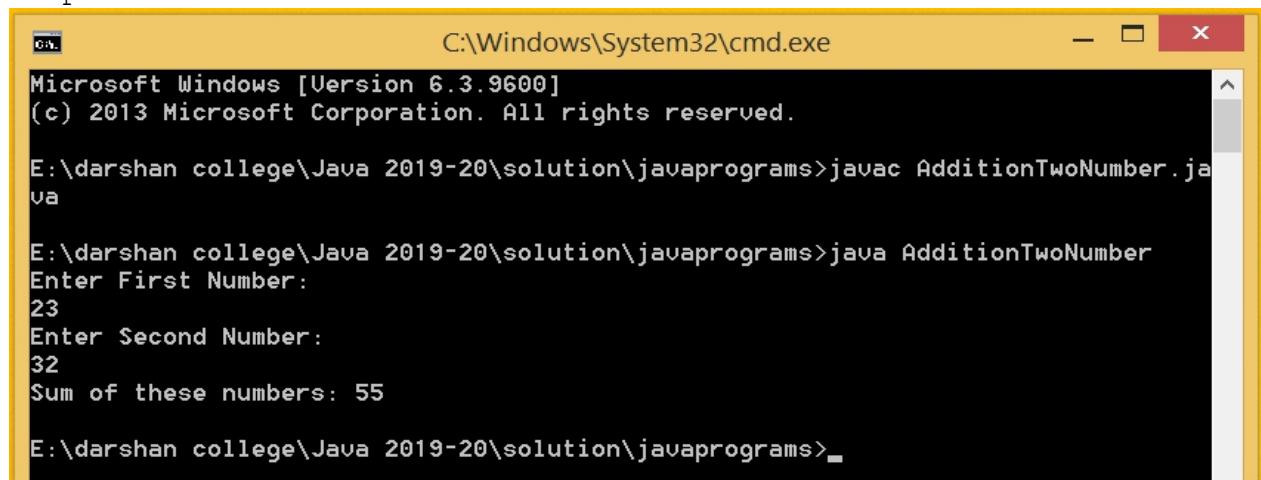
    public static void main(String[] args) {
        //Declare Variables
        int num1, num2, sum;
        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan two integer variable using nextInt()
        System.out.println("Enter First Number: ");
        num1 = sc.nextInt();

        System.out.println("Enter Second Number: ");
        num2 = sc.nextInt();

        sum = num1 + num2;
        System.out.println("Sum of these numbers: "+sum);
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac AdditionTwoNumber.java

E:\darshan college\Java 2019-20\solution\javaprograms>java AdditionTwoNumber
Enter First Number:
23
Enter Second Number:
32
Sum of these numbers: 55

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

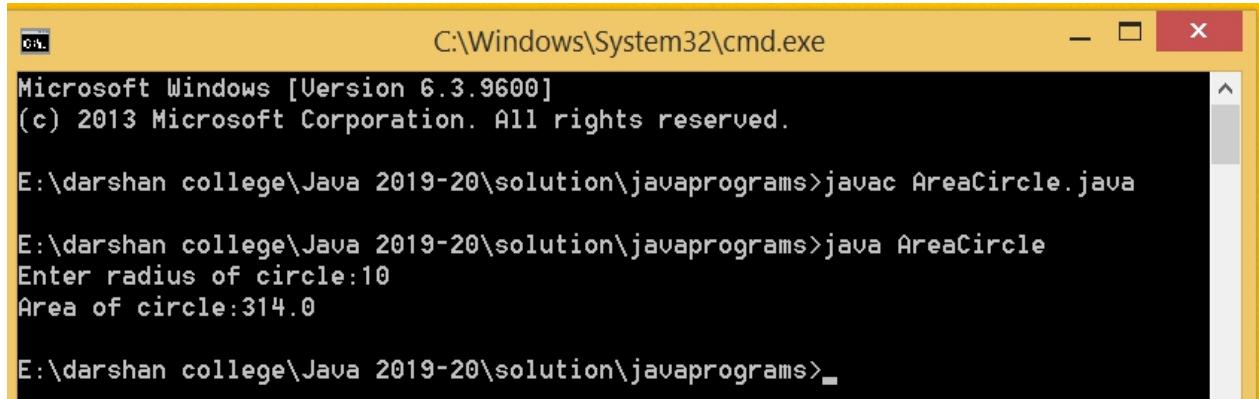
2. WAP to calculate Area of Circle.

```
import java.util.Scanner;
public class AreaCircle {

    public static void main(String[] args) {
        //Declare Variables
        int r;
        double pi = 3.14, area;

        //Declare Scanner
        Scanner s = new Scanner(System.in);
        //Scan Variable
        System.out.print("Enter radius of circle:");
        r = s.nextInt();
        area = pi * r * r;
        System.out.println("Area of circle:"+area);
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac AreaCircle.java
E:\darshan college\Java 2019-20\solution\javaprograms>java AreaCircle
Enter radius of circle:10
Area of circle:314.0

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. WAP to convert temperature from Fahrenheit to Celsius.

```
import java.util.Scanner;

public class FahrenheitToCelsius {

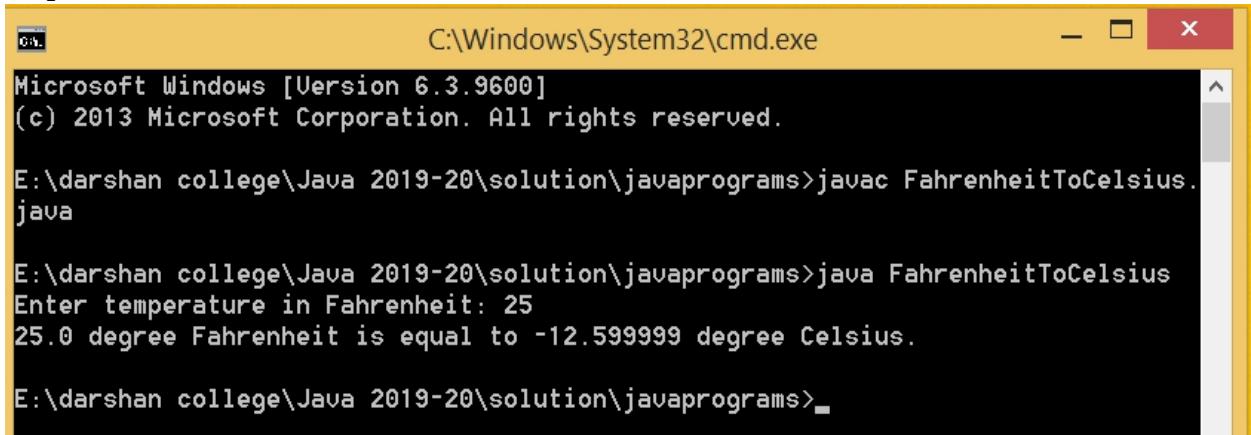
    public static void main(String[] args) {

        //Declare Scanner
        Scanner in = new Scanner(System.in);
```

3140705 – Object Oriented Programming -I

```
// Input temperature in Fahrenheit  
System.out.print("Enter temperature in Fahrenheit: ");  
float F = in.nextFloat();  
  
// Convert Fahrenheit to Celsius  
float C = (F - 32) * (9f / 5);  
  
System.out.println(F + " degree Fahrenheit is equal to " +  
C + " degree Celsius.");  
}  
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following text:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac FahrenheitToCelsius.java

E:\darshan college\Java 2019-20\solution\javaprograms>java FahrenheitToCelsius
Enter temperature in Fahrenheit: 25
25.0 degree Fahrenheit is equal to -12.599999 degree Celsius.

E:\darshan college\Java 2019-20\solution\javaprograms>

4. WAP to find percentage of 5 subjects.

```
import java.util.Scanner;  
  
public class PercentageFiveSub {  
  
    public static void main(String[] args) {  
        //Declare Variables  
        int english, chemistry, computers, physics, maths;  
        double total, Percentage;  
  
        //Declare Scanner  
        Scanner sc = new Scanner(System.in);  
        sc = new Scanner(System.in);  
  
        //Scan 5 subject marks  
        System.out.print("Please Enter the Five Subjects Marks :");
```

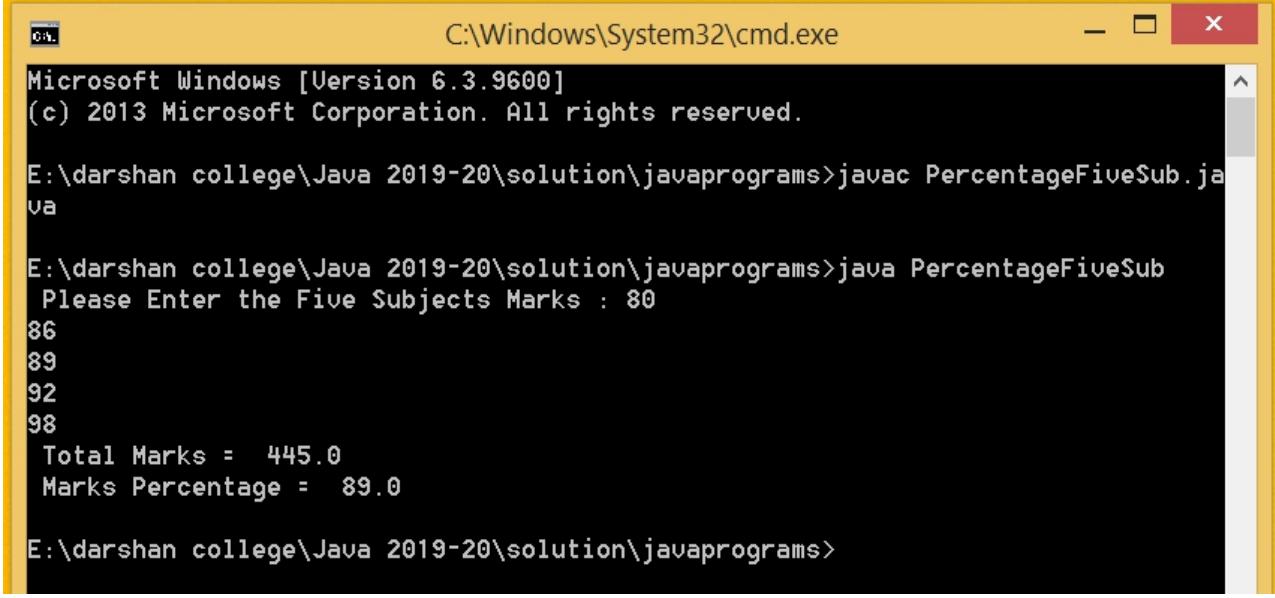
3140705 – Object Oriented Programming -I

```
english = sc.nextInt();
chemistry = sc.nextInt();
computers = sc.nextInt();
physics = sc.nextInt();
maths = sc.nextInt();

total = english + chemistry + computers + physics + maths;
Percentage = (total / 500) * 100;

System.out.println(" Total Marks = " + total);
System.out.println(" Marks Percentage = " + Percentage);
}
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac PercentageFiveSub.java

E:\darshan college\Java 2019-20\solution\javaprograms>java PercentageFiveSub
Please Enter the Five Subjects Marks : 80
86
89
92
98
Total Marks = 445.0
Marks Percentage = 89.0

E:\darshan college\Java 2019-20\solution\javaprograms>
```

5. WAP that reads a number in meters, converts it to feet, and displays the result.

```
import java.util.Scanner;
public class MeterToFeet {

    public static void main(String[] args) {

        //Declare Scanner
        Scanner input = new Scanner(System.in);

        //Scan number in meters
        System.out.print("Enter a value for feet: ");
```

3140705 – Object Oriented Programming -I

```
        double meters = input.nextDouble();

        double feet = meters * 3.281 ;

        System.out.println(meters + " meters is " + feet + " feet");
    }
}
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac MeterToFeet.java

E:\darshan college\Java 2019-20\solution\javaprograms>java MeterToFeet
Enter a value for feet: 3
3.0 meters is 9.843 feet

E:\darshan college\Java 2019-20\solution\javaprograms>
```

6. Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.

Note:- 1 pound=.45359237 Kg and 1 inch=.0254 meters.

```
import java.util.Scanner;
public class BMI {

    public static void main(String[] args) {

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

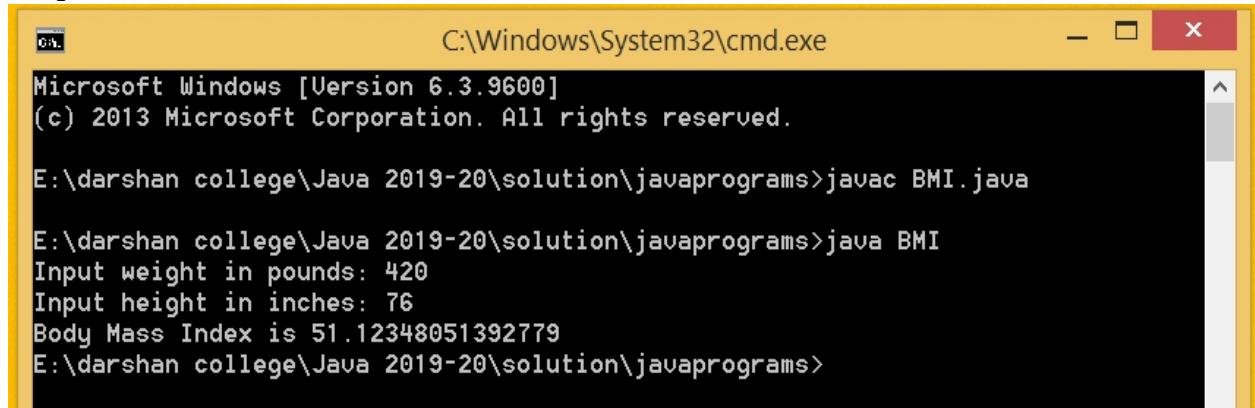
        //Scan weight and height
        System.out.print("Input weight in pounds: ");
        double weight = sc.nextDouble();

        System.out.print("Input height in inches: ");
        double height = sc.nextDouble();
```

3140705 – Object Oriented Programming -I

```
//Convert weight from pounds to kg unit  
double weightKG = weight * 0.45359237;  
  
//Convert height from inches to meter unit  
double heightMeter = height * 0.0254;  
  
//Calculate BMI and print BMI  
double BMI = weightKG/(heightMeter * heightMeter);  
System.out.print("Body Mass Index is " + BMI);  
}  
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
  
E:\darshan college\Java 2019-20\solution\javaprograms>javac BMI.java  
  
E:\darshan college\Java 2019-20\solution\javaprograms>java BMI  
Input weight in pounds: 420  
Input height in inches: 76  
Body Mass Index is 51.12348051392779  
E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-3

1. WAP to check whether the given number is positive or negative.

```
import java.util.Scanner;

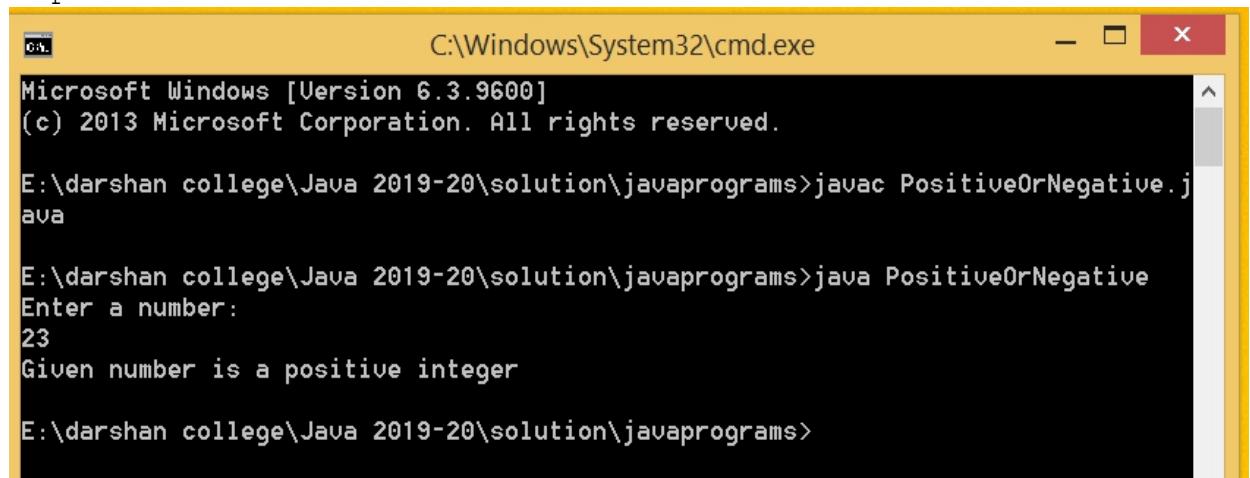
public class PositiveOrNegative {
    public static void main(String args[]) {
        //Declare Variables
        int num;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan number
        System.out.println("Enter a number:");
        num = sc.nextInt();

        if (num > 0)
        {
            System.out.println("Given number is a positive integer");
        }
        else if(num < 0)
        {
            System.out.println("Given number is a negative integer");
        }
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac PositiveOrNegative.java

E:\darshan college\Java 2019-20\solution\javaprograms>java PositiveOrNegative
Enter a number:
23
Given number is a positive integer

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

2. WAP that prompts the user to enter a letter and check whether a letter is a vowel or consonants.

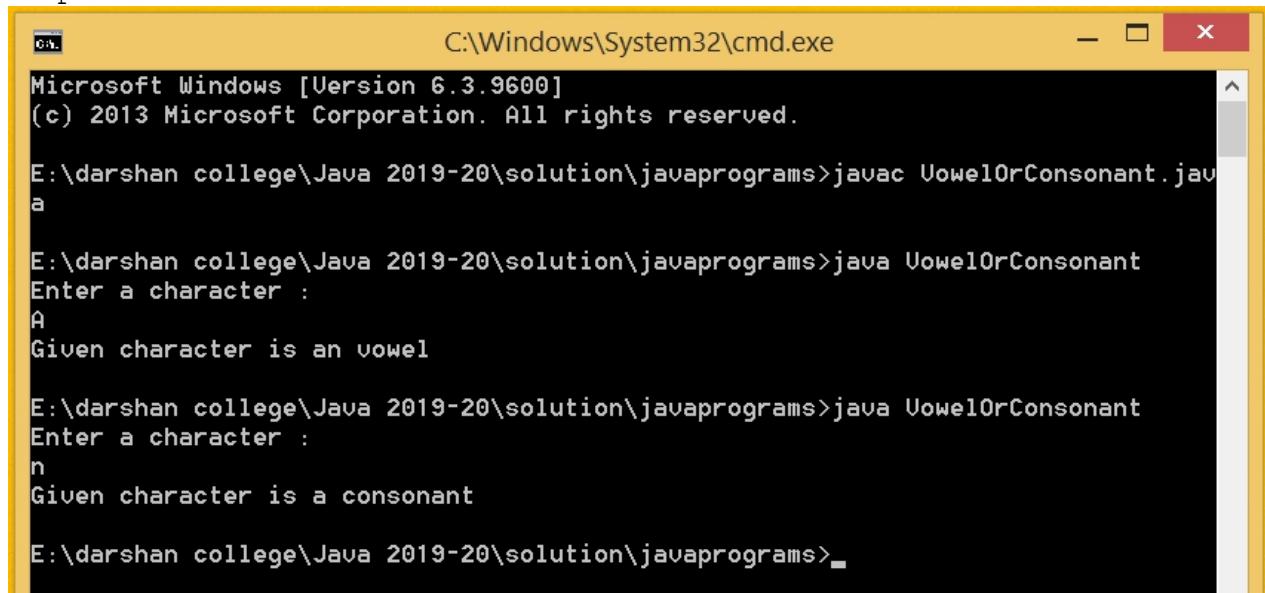
```
import java.util.Scanner;
public class VowelOrConsonant {
    public static void main(String args[]) {

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan letter
        System.out.println("Enter a character :");
        char ch = sc.next().charAt(0);

        if(ch == 'a'|| ch == 'e'|| ch == 'i' ||ch == 'o' ||ch ==
        'u' || ch == 'A'|| ch == 'E'|| ch == 'I' ||ch == 'O' ||ch
        == 'U')
        {
            System.out.println("Given character is an vowel");
        }
        else
        {
            System.out.println("Given character is a consonant");
        }
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac VowelOrConsonant.java

E:\darshan college\Java 2019-20\solution\javaprograms>java VowelOrConsonant
Enter a character :
A
Given character is an vowel

E:\darshan college\Java 2019-20\solution\javaprograms>java VowelOrConsonant
Enter a character :
n
Given character is a consonant

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

3. WAP to find out largest number from given three numbers without using Logical Operat

```
import java.util.Scanner;
public class Largest {

    public static void main(String[] args) {
        //Declare Variables
        int n1, n2, n3;
        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan three integer variable using nextInt()
        System.out.println("Enter First Number: ");
        n1 = sc.nextInt();
        System.out.println("Enter Second Number: ");
        n2 = sc.nextInt();
        System.out.println("Enter Third Number: ");
        n3 = sc.nextInt();

        if(n1 >= n2)
        {
            if(n1 >= n3)
                System.out.println(n1 + " is the largest
number.");
            else
                System.out.println(n3number." is the largest
");
        }
        else
        {   if(n2 >= n3)
            System.out.println(n2number." is the largest
);
            else
                System.out.println(n3number." +
" is the largest
");
        }
    }
}
```

3140705 – Object Oriented Programming -I

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Largest.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Largest
Enter First Number:
23
Enter Second Number:
45
Enter Third Number:
4
45 is the largest number.

E:\darshan college\Java 2019-20\solution\javaprograms>
```

4. WAP to read marks of five subjects. Calculate percentage and print class accordingly. below 35, Pass Class between 35 to 45, Second Class between 45 to 60, First Class between 60 to 70, Distinction if more than 70.

```
import java.util.Scanner;
public class ClassFromPercentage {

    public static void main(String[] args) {
        //Declare Variables
        int english, chemistry, computers, physics, maths;
        double total, percentage;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);
        sc = new Scanner(System.in);

        //Scan 5 subject marks
        System.out.println("Please Enter the Five Subjects
        Marks : ");
        english = sc.nextInt();
        chemistry = sc.nextInt();
        computers = sc.nextInt();
        physics = sc.nextInt();
        maths = sc.nextInt();
        total = english + chemistry + computers + physics +
        maths;
```

3140705 – Object Oriented Programming -I

```
percentage = (total / 500) * 100;

System.out.print("The student class is: ");
if(percentage>=70)
{
    System.out.print("Distinction");
}
else if(percentage>=60 && percentage<70)
{
    System.out.print("First Class");
}
else if(percentage>=45 && percentage<60)
{
    System.out.print("Second Class");
}
else if(percentage>=35 && percentage<45)
{
    System.out.print("Pass Class");
}
else
{
    System.out.print("Fail");
}
}
```

3140705 – Object Oriented Programming -I

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac ClassFromPercentage.java

E:\darshan college\Java 2019-20\solution\javaprograms>java ClassFromPercentage
Please Enter the Five Subjects Marks :
78
89
90
76
77
The student class is: Distinction
E:\darshan college\Java 2019-20\solution\javaprograms>java ClassFromPercentage
Please Enter the Five Subjects Marks :
45
23
12
21
33
The student class is: Fail
E:\darshan college\Java 2019-20\solution\javaprograms>
```

5. WAP to find out largest number from given 3 numbers using conditional operator.

```
import java.util.Scanner;
public class LargestConditional {

    public static void main(String[] args) {
        //Declare Variables
        int n1, n2, n3, largest;
        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan three integer variable using nextInt()
        System.out.println("Enter First Number: ");
        n1 = sc.nextInt();
        System.out.println("Enter Second Number: ");
        n2 = sc.nextInt();
        System.out.println("Enter Third Number: ");
        n3 = sc.nextInt();

        largest = (n1>n2) ? (n1>n3?n1:n3) : (n2>n3?n2:n3);
```

3140705 – Object Oriented Programming -I

```
        System.out.println("Largest Number: "+ largest);
    }
}
```

Output:

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac LargestConditional.j
ava

E:\darshan college\Java 2019-20\solution\javaprograms>java LargestConditional
Enter First Number:
34
Enter Second Number:
32
Enter Third Number:
2
Largest Number: 34

E:\darshan college\Java 2019-20\solution\javaprograms>java LargestConditional
Enter First Number:
76
Enter Second Number:
90
Enter Third Number:
78
Largest Number: 90

E:\darshan college\Java 2019-20\solution\javaprograms>
```

6. WAP to make a Simple Calculator using switch...case

```
import java.util.Scanner;
public class SimpleCalculator {

    public static void main(String[] args) {
        //Declare Variables
        double num1, num2, result;
        char operator;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan two integer variable using nextInt()
```

3140705 – Object Oriented Programming -I

```
System.out.println("Enter First Number: ");
num1 = sc.nextDouble();

System.out.println("Enter Second Number: ");
num2 = sc.nextDouble();

System.out.print("Enter an operator (+, -, *, /): ");
operator = sc.next().charAt(0);

switch(operator)
{
    case '+':
        result = num1 + num2;
        break;

    case '-':
        result = num1 - num2;
        break;

    case '*':
        result = num1 * num2;
        break;

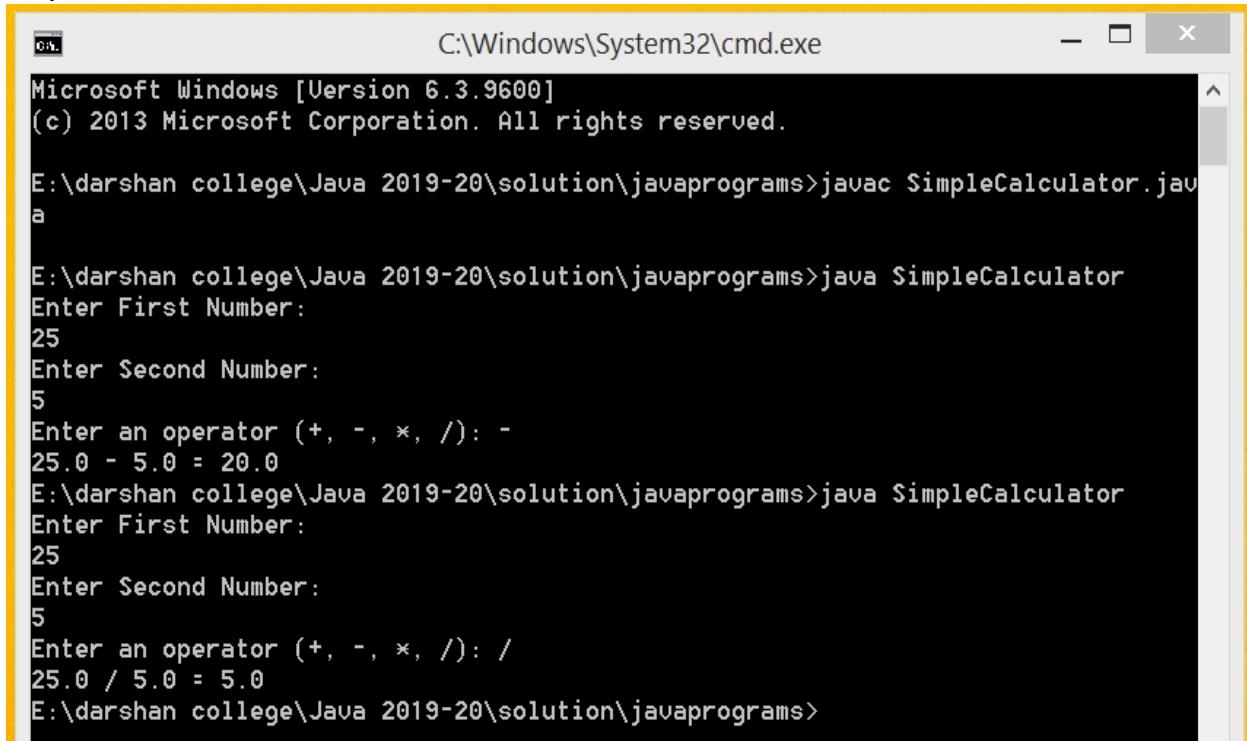
    case '/':
        result = num1 / num2;
        break;

    default:
        System.out.printf("Operator is not correct");
        return;
}

System.out.printf(num1 + " " + operator + " " + num2 + " = "
"+result);
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac SimpleCalculator.java

E:\darshan college\Java 2019-20\solution\javaprograms>java SimpleCalculator
Enter First Number:
25
Enter Second Number:
5
Enter an operator (+, -, *, /): -
25.0 - 5.0 = 20.0
E:\darshan college\Java 2019-20\solution\javaprograms>java SimpleCalculator
Enter First Number:
25
Enter Second Number:
5
Enter an operator (+, -, *, /): /
25.0 / 5.0 = 5.0
E:\darshan college\Java 2019-20\solution\javaprograms>
```

7. Three sides of a triangle are entered through the keyboard. WAP to check whether triangle is isosceles, equilateral, scalene or right angled triangle.

```
import java.util.Scanner;
public class TriangleType {

    public static void main(String[] args) {
        //Declare Variables
        int a,b,c;

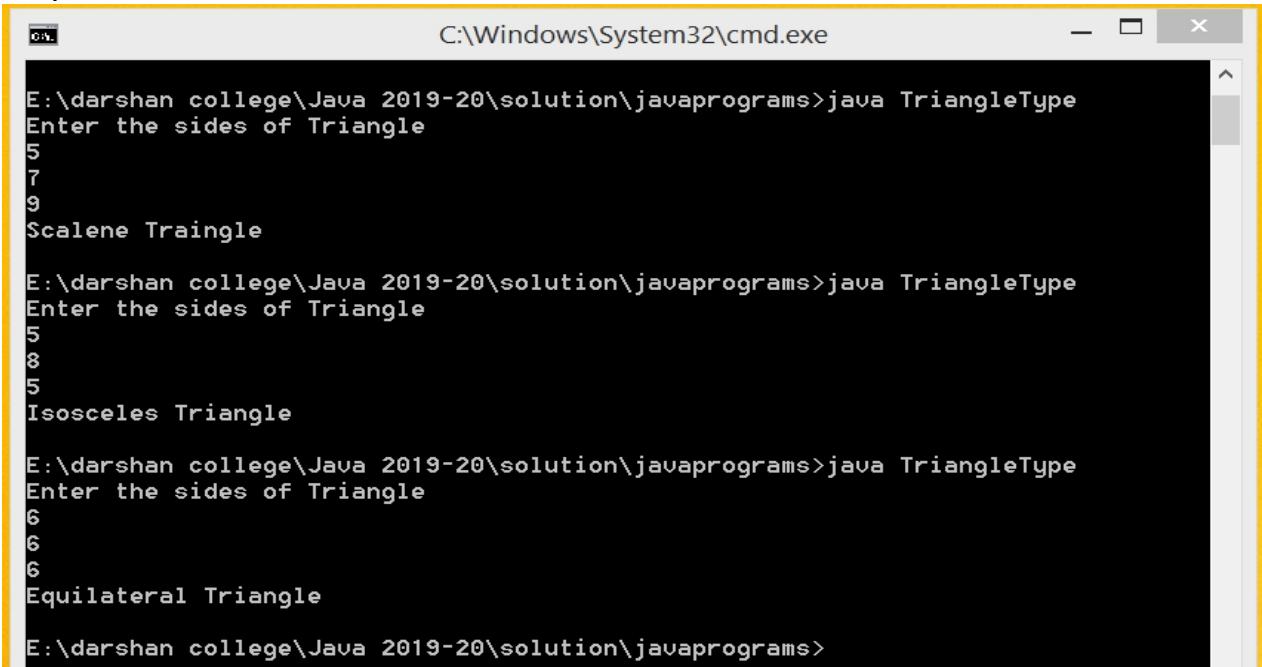
        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan three sides of triangle
        System.out.println("Enter the sides of Triangle");
        a=sc.nextInt();
        b=sc.nextInt();
        c=sc.nextInt();

        if(a==b&&b==c)
            System.out.println("Equilateral Triangle");
```

3140705 – Object Oriented Programming -I

```
        else if(a*a+b*b==c*c || b*b+c*c==a*a || a*a+c*c==b*b)
            System.out.println("Right Angle Triangle");
        else if(a==b||b==c||c==a)
            System.out.println("Isosceles Triangle");
        else
            System.out.println("Scalene Traingle");
    }
}
```

Output:

```
C:\Windows\System32\cmd.exe
E:\darshan college\Java 2019-20\solution\javaprograms>java TriangleType
Enter the sides of Triangle
5
7
9
Scalene Traingle

E:\darshan college\Java 2019-20\solution\javaprograms>java TriangleType
Enter the sides of Triangle
5
8
5
Isosceles Triangle

E:\darshan college\Java 2019-20\solution\javaprograms>java TriangleType
Enter the sides of Triangle
6
6
6
Equilateral Triangle

E:\darshan college\Java 2019-20\solution\javaprograms>
```

8. WAP that prompts the user to input number of calls and calculate the monthly telephone bills as per the following rule:
- Minimum Rs. 200 for up to 100 calls.
- Plus Rs. 0.60 per call for next 50 calls.
- Plus Rs. 0.50 per call for next 50 calls.
- Plus Rs. 0.40 per call for any call beyond 200 calls.

```
import java.util.Scanner;
public class TelephoneBill {
    public static void main(String args[]) {
        //Declare Variables
        int totalCalls;
        double billAmount;
```

3140705 – Object Oriented Programming -I

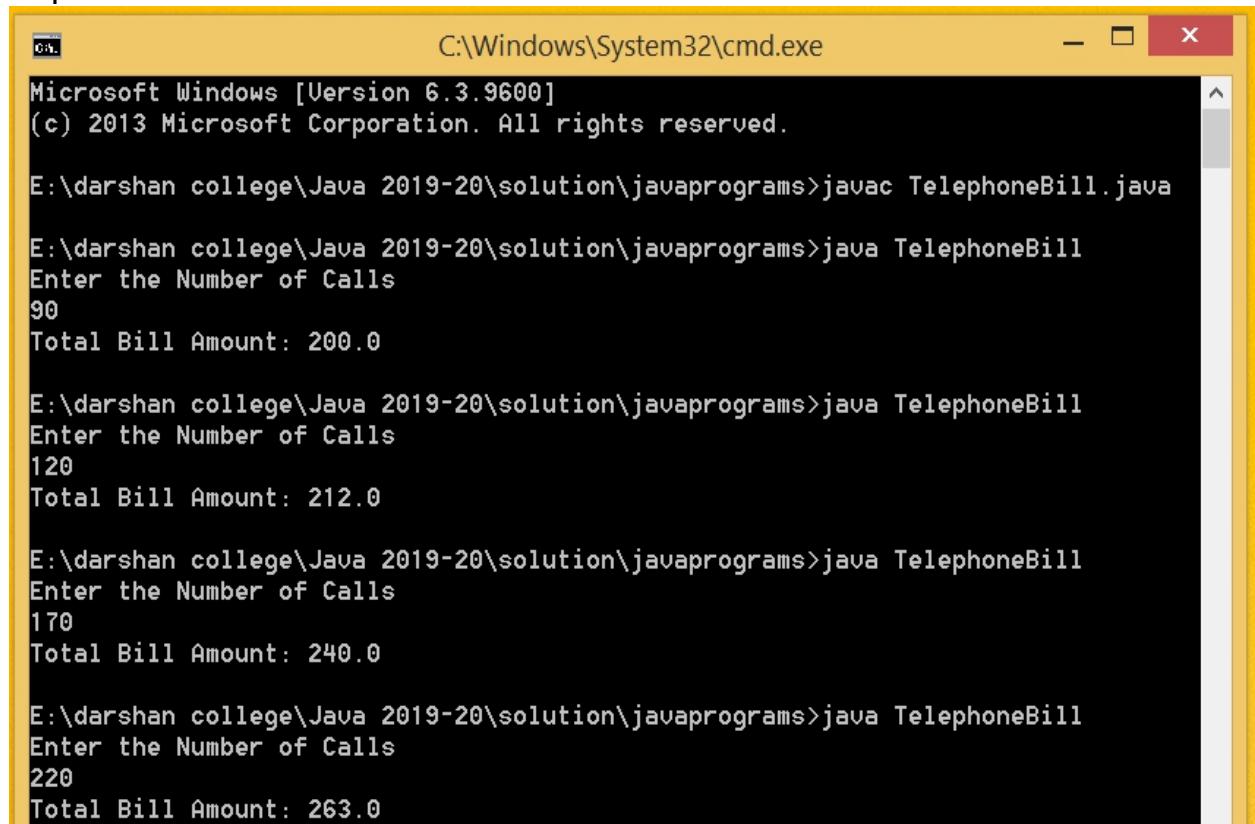
```
//Declare Scanner
Scanner sc = new Scanner(System.in);

//Scan number
System.out.println("Enter the Number of Calls");
totalCalls = sc.nextInt();

if (totalCalls<=100)
    billAmount=200;
else if (totalCalls>100 && totalCalls<=150)
    billAmount=200+(0.60*(totalCalls-100));
else if (totalCalls>150 && totalCalls<=200)
    billAmount=200+(0.60*50)+(0.50*(totalCalls-150));
else
    billAmount=200+(0.60*50)+(0.50*50)+(0.40*(totalCalls-
200));

System.out.println("Total Bill Amount: " + billAmount);
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac TelephoneBill.java

E:\darshan college\Java 2019-20\solution\javaprograms>java TelephoneBill
Enter the Number of Calls
90
Total Bill Amount: 200.0

E:\darshan college\Java 2019-20\solution\javaprograms>java TelephoneBill
Enter the Number of Calls
120
Total Bill Amount: 212.0

E:\darshan college\Java 2019-20\solution\javaprograms>java TelephoneBill
Enter the Number of Calls
170
Total Bill Amount: 240.0

E:\darshan college\Java 2019-20\solution\javaprograms>java TelephoneBill
Enter the Number of Calls
220
Total Bill Amount: 263.0
```

3140705 – Object Oriented Programming -I

Practical-4

1. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3.

```
import java.util.Scanner;
public class DivisibleBy2Not3 {

    public static void main(String[] args) {
        //Declare Variables
        int num1, num2;
        //Declare Scanner
        Scanner sc = new Scanner(System.in);

        //Scan two integer variable
        System.out.println("Enter First Number: ");
        num1 = sc.nextInt();

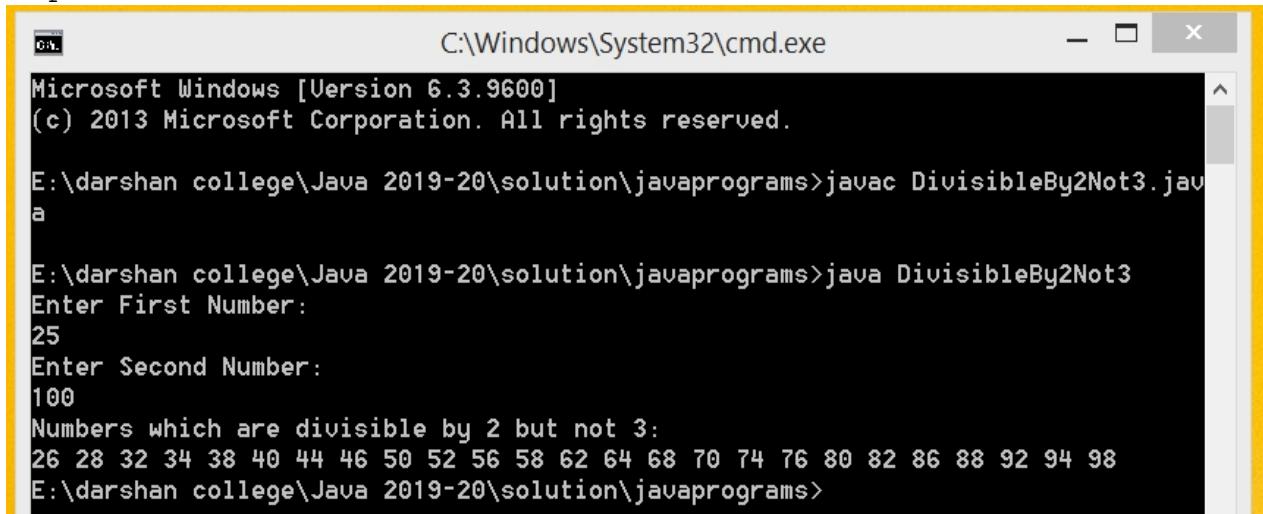
        System.out.println("Enter Second Number: ");
        num2 = sc.nextInt();

        System.out.println("Numbers which are divisible by 2 but
not 3:");

        // iterate from num1 to num2
        for (int i = num1; i < num2; i++)
        {
            if (i % 2 == 0 && i % 3 != 0)
                System.out.print(i + " ");
        }
    }
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac DivisibleBy2Not3.java

E:\darshan college\Java 2019-20\solution\javaprograms>java DivisibleBy2Not3
Enter First Number:
25
Enter Second Number:
100
Numbers which are divisible by 2 but not 3:
26 28 32 34 38 40 44 46 50 52 56 58 62 64 68 70 74 76 80 82 86 88 92 94 98
E:\darshan college\Java 2019-20\solution\javaprograms>
```

2. WAP to find factorial of the given number.

```
import java.util.Scanner;
public class Factorial {

    public static void main(String[] args) {
        //Declare Variables
        int number,i,fact=1;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

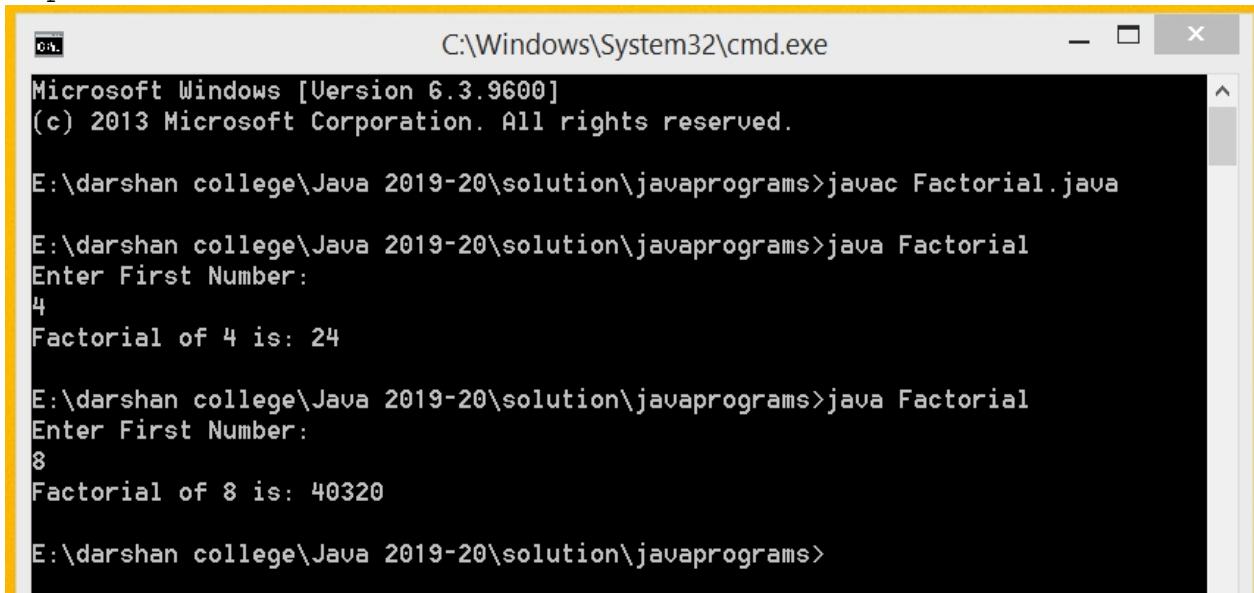
        //Scan integer variable
        System.out.println("Enter First Number: ");
        number = sc.nextInt();

        // iterate from 1 to number
        for(i=1;i<=number;i++)
        {
            fact=fact*i;
        }

        System.out.println("Factorial of "+number+" is: "+fact);
    }
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Factorial.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Factorial
Enter First Number:
4
Factorial of 4 is: 24

E:\darshan college\Java 2019-20\solution\javaprograms>java Factorial
Enter First Number:
8
Factorial of 8 is: 40320

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. WAP to find whether the given number is prime or not.

```
import java.util.*;
class PrimeSimple{
    public static void main(String[] args)
        {int n, prime=0;
        Scanner sc = new Scanner(System.in);

        //Get details
        System.out.print("Enter Number : ");
        n = sc.nextInt();

        for(int i = 2; i < n;
            i++) {if(n % i == 0)
            {
                prime = 1;
                break;
            }
        }

        if(prime == 1)
            System.out.println("Number is not Prime");
        else
            System.out.println("Number is Prime");
    }
}
```

3140705 – Object Oriented Programming -I

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac PrimeSimple.java

E:\darshan college\Java 2019-20\solution\javaprograms>java PrimeSimple
Enter Number : 4
Number is not Prime

E:\darshan college\Java 2019-20\solution\javaprograms>java PrimeSimple
Enter Number : 7
Number is Prime

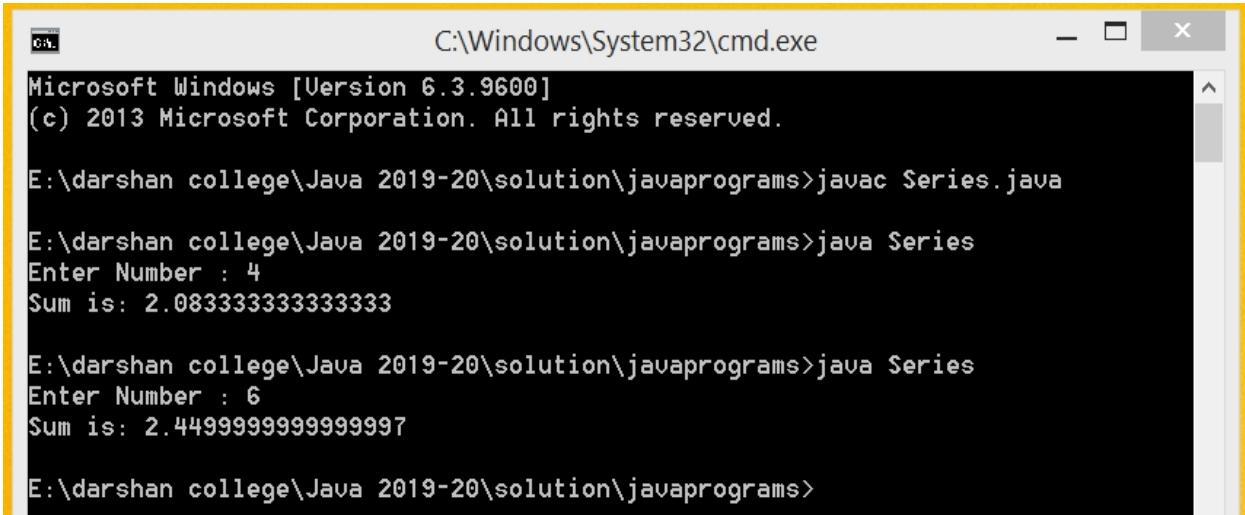
E:\darshan college\Java 2019-20\solution\javaprograms>
```



```
import java.util.*;  
  
class Series{  
    public static void main(String[] args) {  
        //Declare Variables  
        int n;  
        double sum=0;  
        //Declare Scanner  
        Scanner sc = new Scanner(System.in);  
  
        //Scan variable  
        System.out.print("Enter Number : ");  
        n = sc.nextInt();  
  
        for (int i = 1; i <= n; i++)  
            sum = sum + 1.0/i;  
  
        System.out.println("Sum is: "+ sum);  
    }  
}
```

3140705 – Object Oriented Programming -I

Output:



A screenshot of a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window shows the following text:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
E:\darshan college\Java 2019-20\solution\javaprograms>javac Series.java
E:\darshan college\Java 2019-20\solution\javaprograms>java Series
Enter Number : 4
Sum is: 2.083333333333333
E:\darshan college\Java 2019-20\solution\javaprograms>java Series
Enter Number : 6
Sum is: 2.4499999999999997
E:\darshan college\Java 2019-20\solution\javaprograms>

5. WAP to print given number in reverse order.

```
import java.util.Scanner;
public class ReverseNumber {

    public static void main(String[] args) {
        //Declare Variables
        int number,i,reversed = 0;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

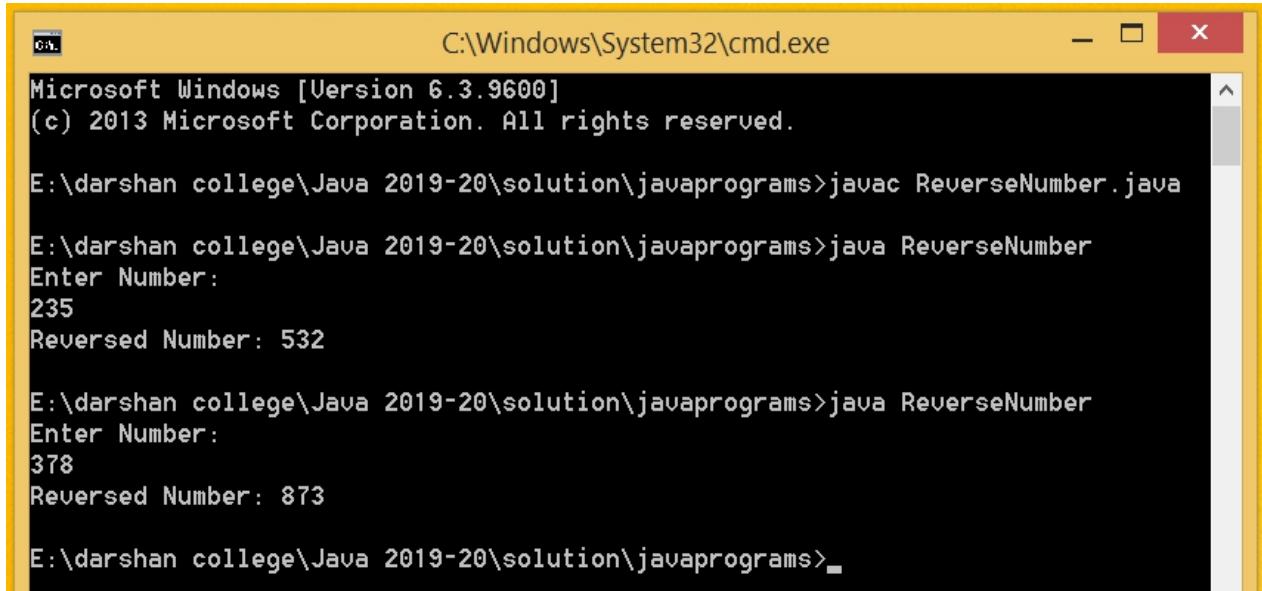
        //Scan integer variable
        System.out.println("Enter Number: ");
        number = sc.nextInt();

        while(number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
        }

        System.out.println("Reversed Number: " + reversed);
    }
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac ReverseNumber.java
E:\darshan college\Java 2019-20\solution\javaprograms>java ReverseNumber
Enter Number:
235
Reversed Number: 532

E:\darshan college\Java 2019-20\solution\javaprograms>java ReverseNumber
Enter Number:
378
Reversed Number: 873

E:\darshan college\Java 2019-20\solution\javaprograms>
```

6. WAP program to calculate the sum of all positive even numbers and the sum of all negative odd numbers from a set of numbers. You can enter 0 (zero) to quit the program and the program displays the result.

```
import java.util.Scanner;
public class SumOddEven {

    public static void main(String[] args) {
        //Declare Variables
        int sumOdd=0,sumEven=0,number;

        //Declare Scanner
        Scanner sc = new Scanner(System.in);

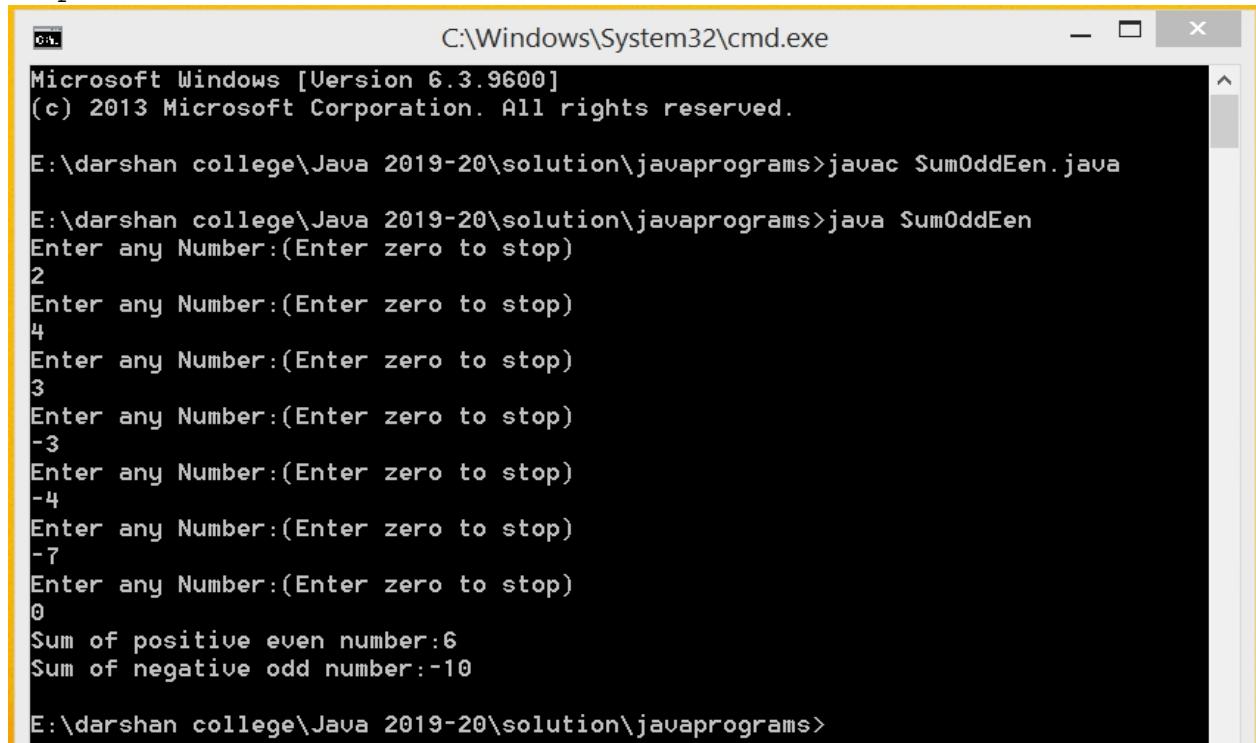
        while(true)
        {
            //Scan integer variable, enter zero to stop
            System.out.println("Enter any Number: (Enter zero to stop)");
            number = sc.nextInt();

            if(number>0 && number % 2 == 0)
                sumEven+=number;
            if(number<0 && number % 2 != 0)
                sumOdd+=number;
            if(number==0)
```

3140705 – Object Oriented Programming -I

```
        break;
    }
    System.out.println("Sum of positive even number:" + sumEven);
    System.out.println("Sum of negative odd number:" + sumOdd);
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac SumOddEen.java

E:\darshan college\Java 2019-20\solution\javaprograms>java SumOddEen
Enter any Number:(Enter zero to stop)
2
Enter any Number:(Enter zero to stop)
4
Enter any Number:(Enter zero to stop)
3
Enter any Number:(Enter zero to stop)
-3
Enter any Number:(Enter zero to stop)
-4
Enter any Number:(Enter zero to stop)
-7
Enter any Number:(Enter zero to stop)
0
Sum of positive even number:6
Sum of negative odd number:-10

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

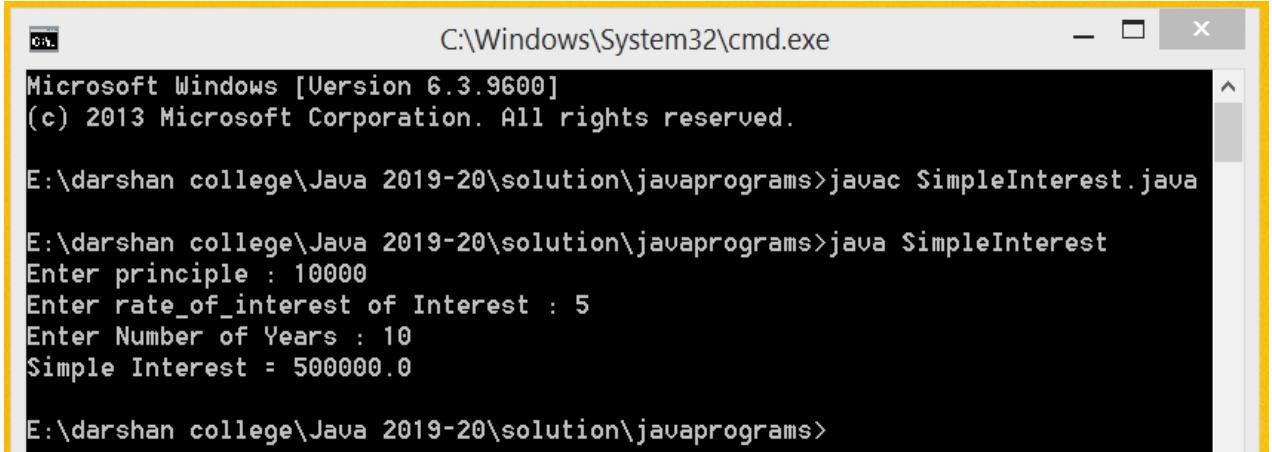
Practical-5

1. WAP to calculate simple interest using method

```
import java.util.*;  
  
class SimpleInterest{  
    //Method  
    static void calculateInterest(double principle, double rate_of_interest, double number_of_years){  
        double simple_interest;  
        //Calculate Interest & Print  
        simple_interest = principle * rate_of_interest * number_of_years;  
        System.out.println("Simple Interest = " + simple_interest);  
    }  
  
    //Main Method  
    public static void main(String[] args) {  
        double principle, rate_of_interest, number_of_years;  
        Scanner sc = new Scanner(System.in);  
  
        //Get details  
        System.out.print("Enter principle : ");  
        principle = sc.nextDouble();  
        System.out.print("Enter rate_of_interest of Interest : ");  
        rate_of_interest = sc.nextDouble();  
        System.out.print("Enter Number of Years : ");  
        number_of_years = sc.nextDouble();  
  
        //Method call  
        calculateInterest(principle,rate_of_interest,  
        number_of_years);  
    }  
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The output of the program is displayed:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac SimpleInterest.java

E:\darshan college\Java 2019-20\solution\javaprograms>java SimpleInterest
Enter principle : 10000
Enter rate_of_interest of Interest : 5
Enter Number of Years : 10
Simple Interest = 500000.0

E:\darshan college\Java 2019-20\solution\javaprograms>
```

2. WAP to find maximum number from given two numbers using method.

```
import java.util.*;

class Max_2{
    //Method
    static void maxNumber(int number1, int
        number2){int max = 0, equal = 0;

        //Find Max
        if(number1 > number2)
            max = number1;
        else if(number2 > number1)
            max = number2;
        else
            equal = 1;

        //Print Max
        if(equal == 0)
            System.out.println("Max number = " + max);
        else
            System.out.println("Both number are equal");
    }

    //Main Method
    public static void main(String[] args)
        {int number1, number2;
        Scanner sc = new Scanner(System.in);

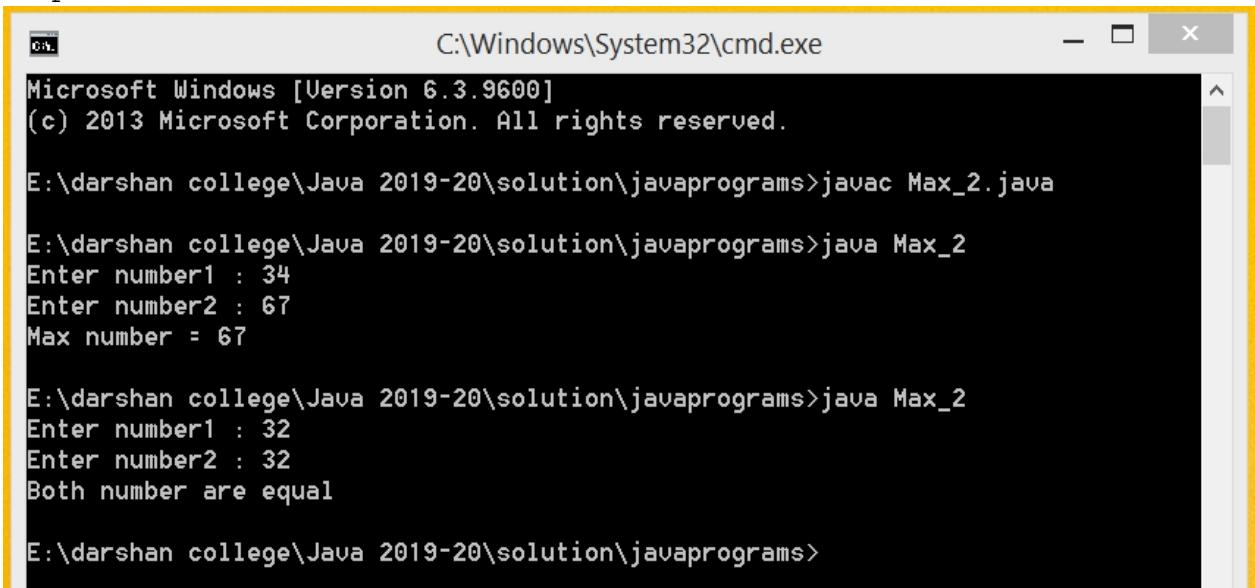
        //Get details
```

3140705 – Object Oriented Programming -I

```
System.out.print("Enter number1 : ");
number1 = sc.nextInt();
System.out.print("Enter number2 : ");
number2 = sc.nextInt();

//Method call
maxNumber(number1, number2);
}
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Max_2.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Max_2
Enter number1 : 34
Enter number2 : 67
Max number = 67

E:\darshan college\Java 2019-20\solution\javaprograms>java Max_2
Enter number1 : 32
Enter number2 : 32
Both number are equal

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. WAP to generate Fibonacci series of N given number using method.

```
import java.util.*;

class Fibonacci{
    //Method
    static void printFibonacci(int
        n){int n1 = 1, n2 = 1, temp;

        for(int count = 0 ; count < n;
            count++){System.out.println(n1);
            temp = n2;
            n2 = n1 + n2;
            n1 = temp;
        }
    }
}
```

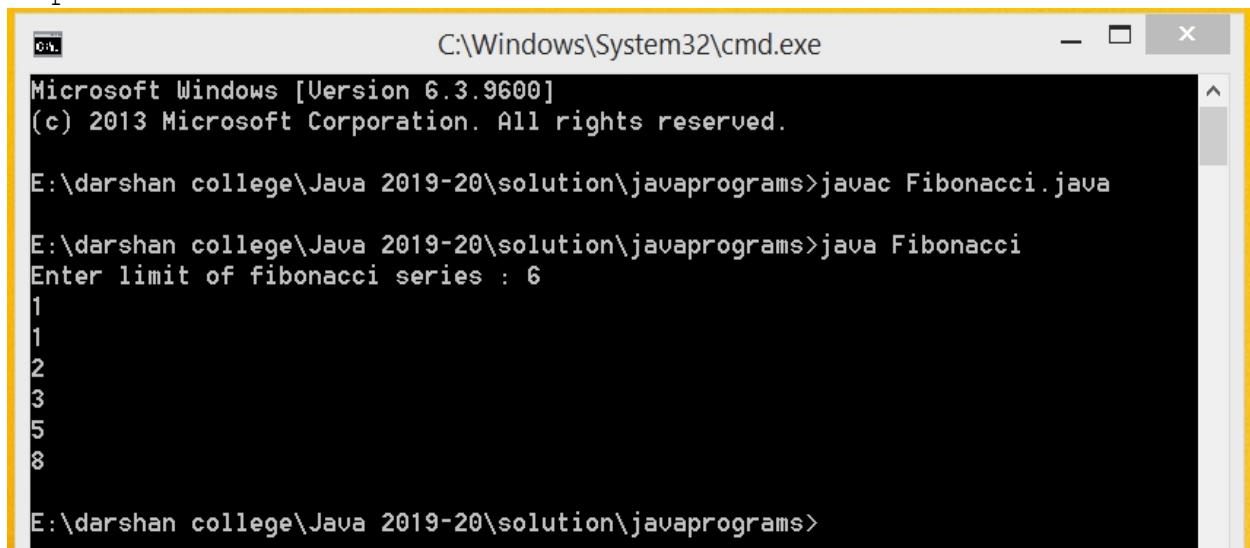
3140705 – Object Oriented Programming -I

```
//Main Method
public static void main(String[] args)
{
    int n;
    Scanner sc = new Scanner(System.in);

    //Get details
    System.out.print("Enter limit of fibonacci series : ");
    n = sc.nextInt();

    //Method call
    printFibonacci(n);
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'cmd.exe' with the path 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Fibonacci.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Fibonacci
Enter limit of fibonacci series : 6
1
1
2
3
5
8

E:\darshan college\Java 2019-20\solution\javaprograms>
```

4. WAP to accept a number and check whether the number is prime or not. Use `method` name `check (int n)`. The method returns 1, if the number is prime otherwise, it returns 0.

```
import java.util.*;

class Prime{
    //Method
    static int check(int n){
        for(int i = 2; i < n;
            i++){if(n % i == 0)
                return 0;
    }
}
```

3140705 – Object Oriented Programming -I

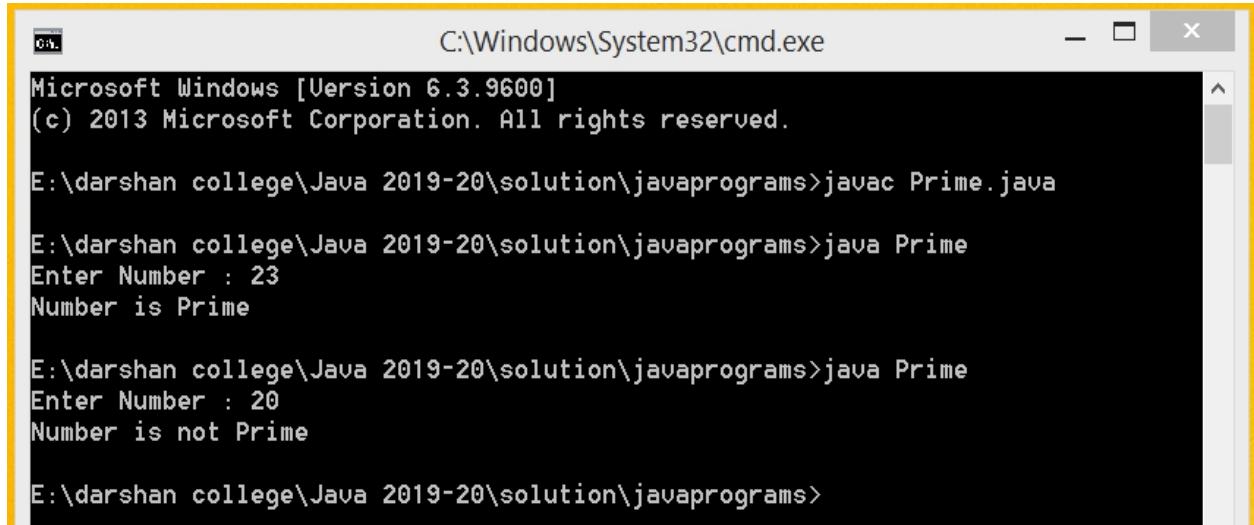
```
        return 1;
    }

//Main Method
public static void main(String[] args)
    {int n, prime;
    Scanner sc = new Scanner(System.in);

    //Get details
    System.out.print("Enter Number : ");
    n = sc.nextInt();

    //Method call
    prime = check(n);
    if(prime == 0)
        System.out.println("Number is not Prime");
    else
        System.out.println("Number is Prime");
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Prime.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Prime
Enter Number : 23
Number is Prime

E:\darshan college\Java 2019-20\solution\javaprograms>java Prime
Enter Number : 20
Number is not Prime

E:\darshan college\Java 2019-20\solution\javaprograms>
```

5. WAP that calculates area of circle, triangle and square using method overloading.

```
import java.util.*;

class AreaOverloading{
    //Method
    void area(double radius) {
```

3140705 – Object Oriented Programming -I

```
        double aoc;
        aoc = 3.14 * radius * radius;
        System.out.println("Area of Circle : "+ aoc);
    }
    void area(double base, double
              altitude){double aot;
              aot = 0.5 * base * altitude;
              System.out.println("Area of Triangle : "+ aot);
    }
    void area(float
              side){double aos;
              aos = side * side;
              System.out.println("Area of Square : "+ aos);
    }

//Main Method
public static void main(String[] args)
{
    int ch;
    double radius, base, altitude;
    float side;
    Scanner sc = new Scanner(System.in);

    AreaOverloading ao = new AreaOverloading();

    //Get details
    System.out.print("Operations:-\n1.Area of Circle\n2.Area of
                     Triangle\n3.Area of Square\nEnter choice : ");
    ch = sc.nextInt();
    switch(ch) {
        case 1:
            System.out.print("Enter radius : ");
            radius = sc.nextDouble();
            ao.area(radius);
            break;
        case 2:
            System.out.print("Enter base length : ");
            base = sc.nextDouble();
            System.out.print("Enter altitude : ");
            altitude = sc.nextDouble();
            ao.area(base,altitude);
            break;
        case 3:
            System.out.print("Enter side length : ");
            side = sc.nextFloat();
            ao.area(side);
    }
}
```

3140705 – Object Oriented Programming -I

```
        break;
    default:
        System.out.println("Invalid choice!");
    }
}
}
```

Output:

```
C:\Windows\System32\cmd.exe
E:\darshan college\Java 2019-20\solution\javaprograms>java AreaOverloading
Operations:-
1.Area of Circle
2.Area of Triangle
3.Area of Square
Enter choice : 1
Enter radius : 20
Area of Circle : 1256.0

E:\darshan college\Java 2019-20\solution\javaprograms>java AreaOverloading
Operations:-
1.Area of Circle
2.Area of Triangle
3.Area of Square
Enter choice : 2
Enter base length : 22
Enter altitude : 32
Area of Triangle : 352.0

E:\darshan college\Java 2019-20\solution\javaprograms>java AreaOverloading
Operations:-
1.Area of Circle
2.Area of Triangle
3.Area of Square
Enter choice : 3
```

6. Write a method with following method header: public int gcd (int num1, int num2). Write a program that prompts the user to enter two integers and compute the gcd of two integers. [Note: The greatest common divisor (GCD) of two numbers is the largest number that divides them both.]

```
import java.util.*;
class GCD{
    //Method
    static void gcd(int num1, int
        num2){int ans = 1, flag = 0;
```

3140705 – Object Oriented Programming -I

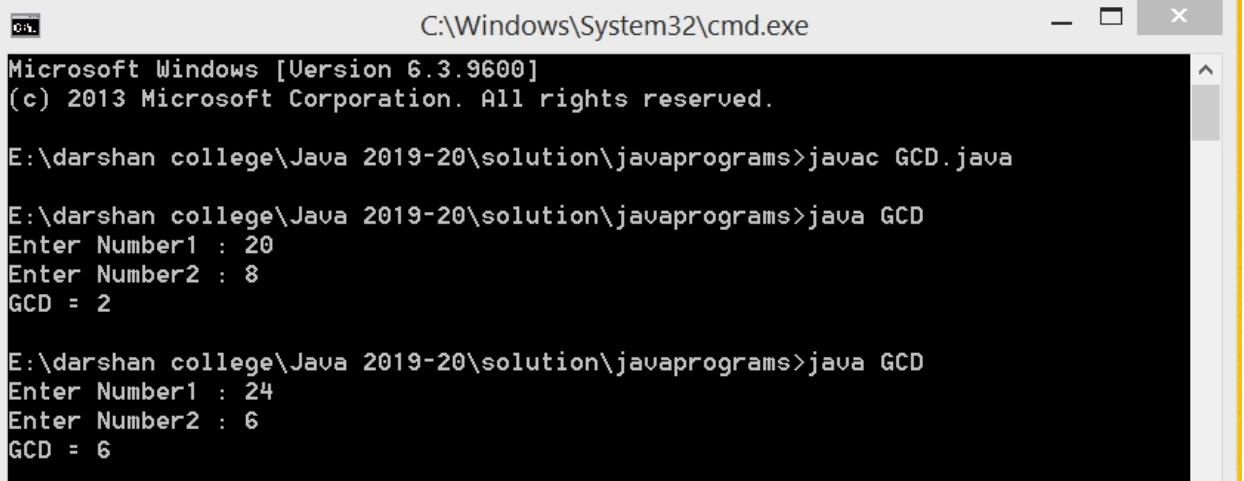
```
for(int i = 2; i<=num1 || i<=num2;
    i++) {for(int j = 2; j < i; j++) {
        if(i % j == 0)
            flag = 1;
    }
    if(flag == 0 && num1 % i == 0 && num2 % i ==
        0) {ans = ans * i;
    }
    flag = 0;
}
System.out.println("GCD = "+ ans);
}

//Main Method
public static void main(String[] args)
{
    int num1, num2;
    Scanner sc = new Scanner(System.in);

    //Get details
    System.out.print("Enter Number1 : ");
    num1 = sc.nextInt();
    System.out.print("Enter Number2 : ");
    num2 = sc.nextInt();

    //Method call
    gcd(num1, num2);
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac GCD.java

E:\darshan college\Java 2019-20\solution\javaprograms>java GCD
Enter Number1 : 20
Enter Number2 : 8
GCD = 2

E:\darshan college\Java 2019-20\solution\javaprograms>java GCD
Enter Number1 : 24
Enter Number2 : 6
GCD = 6
```

3140705 – Object Oriented Programming -I

Practical-6

1. WAP to count number of even or odd number from an array of n numbers.

```
import java.util.*;
class CountOddEven {
    public static void main(String[] args)
        {int a[], n, odd=0, even=0;
        Scanner sc = new Scanner(System.in);
        //Get size of array
        System.out.print("Enter size of array : ");
        n = sc.nextInt();
        a = new int[n];
        //Get elements of array
        for(int i = 0; i < a.length; i++)
            { System.out.print("Enter number :
                ");a[i] = sc.nextInt();
            }
        //Count number of odd and even numbers
        for(int i = 0; i < a.length; i++) {
            if(a[i]%2==0)
                even++;
            else
                odd++;
            }
        //Displaying result
        System.out.println("\nTotal odd: " + odd);
        System.out.println("\nTotal even: " + even);
    }
}
```

3140705 – Object Oriented Programming -I

Output:

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The output displays the standard Windows copyright message, followed by the command 'javac CountOddEven.java' being run from the directory 'E:\darshan college\Java 2019-20\solution\javaprograms'. The program then prompts for the size of an array (5) and five numbers (23, 45, 66, 88, 90). It outputs the counts: 'Total odd: 2' and 'Total even: 3'.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac CountOddEven.java

E:\darshan college\Java 2019-20\solution\javaprograms>java CountOddEven
Enter size of array : 5
Enter number : 23
Enter number : 45
Enter number : 66
Enter number : 88
Enter number : 90

Total odd: 2

Total even: 3
```

2. WAP to accept n numbers in an array. Display the sum of all the numbers which divisible by either 3 or 5.

```
import java.util.*;
class Div_3_5{
    public static void main(String[] args)
        {int a[], n, sum = 0;
        Scanner sc = new Scanner(System.in);
        //Get size of array
        System.out.print("Enter size of array : ");
        n = sc.nextInt();
        a = new int[n];
        //Get elements of array
        for(int i = 0; i < a.length;
            i++){ System.out.print("Enter
        number : ");a[i] = sc.nextInt();

        //Add numbers divisible by 3 or 5
        if(a[i] % 3 == 0 || a[i] % 5 == 0){
            sum += a[i];
        }
    }
    System.out.println("Sum is "+ sum);
}
```

3140705 – Object Oriented Programming -I

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac Div_3_5.java

E:\darshan college\Java 2019-20\solution\javaprograms>java Div_3_5
Enter size of array : 5
Enter number : 20
Enter number : 21
Enter number : 34
Enter number : 55
Enter number : 90
Sum is 186

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. WAP to accept n numbers in an array. Now, enter a number and search whether the number is present or not in the list of array elements by using linear search.

```
import java.util.*;
class ArraySearch{
    public static void main(String[] args)
        {int a[], n, x;
        String message = "not found";
        Scanner sc = new Scanner(System.in);
        //Get size of array
        System.out.print("Enter size of array : ");
        n = sc.nextInt();
        a = new int[n];
        //Get elements of array
        for(int i = 0; i < a.length;
            i++){ System.out.print("Enter
                number : ");a[i] = sc.nextInt();
        }
        System.out.print("Enter number to search : ");
        x = sc.nextInt();
        for(int i = 0; i < a.length;
            i++){if(a[i] == x){
                message = "found at "+(i+1);
                break;
            }
        }
    }
}
```

3140705 – Object Oriented Programming -I

```
        System.out.println("Element "+message);
    }
}
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac ArraySearch.java

E:\darshan college\Java 2019-20\solution\javaprograms>java ArraySearch
Enter size of array : 5
Enter number : 22
Enter number : 44
Enter number : 56
Enter number : 78
Enter number : 90
Enter number to search : 78
Element found at 4

E:\darshan college\Java 2019-20\solution\javaprograms>java ArraySearch
Enter size of array : 3
Enter number : 22
Enter number : 33
Enter number : 44
Enter number to search : 67
Element not found
```

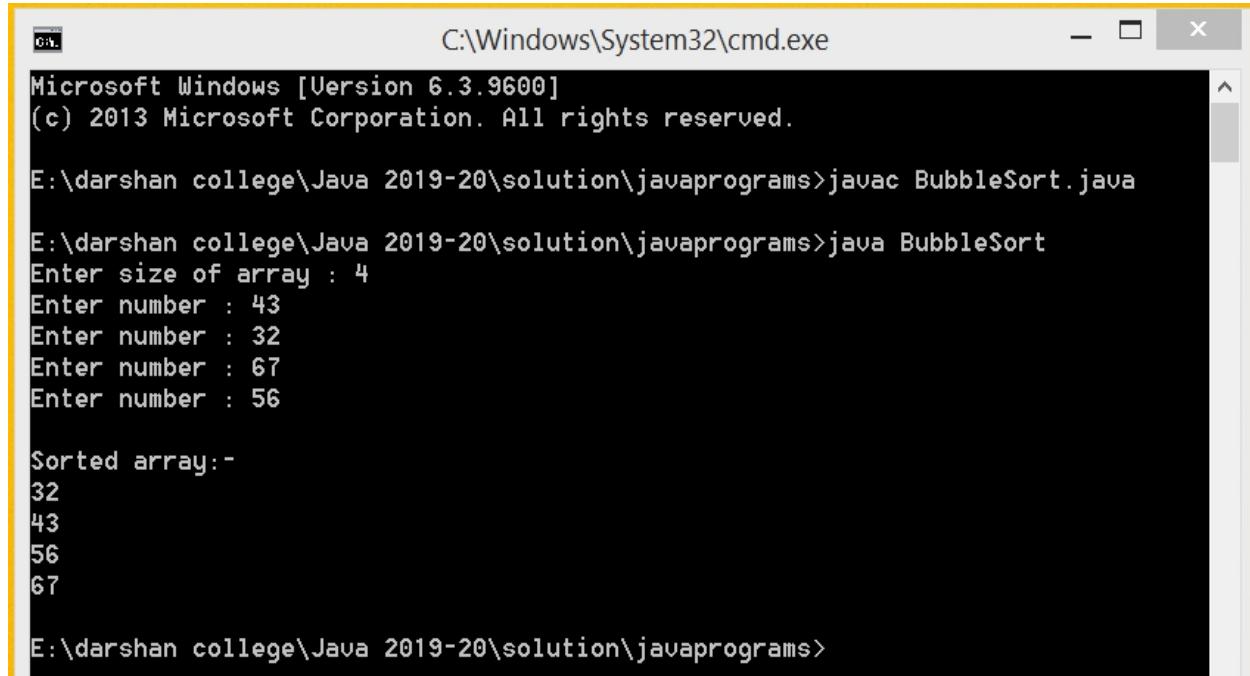
4. WAP to accept 10 numbers in an array. Pass this array to a function name bubble_s (int m []). Arrange all the numbers in ascending order using bubble sort and display the

```
import java.util.*;
class BubbleSort {
    public static void main(String[] args)
    {
        int a[], n, temp;
        Scanner sc = new Scanner(System.in);
        //Get size of array
        System.out.print("Enter size of array : ");
        n = sc.nextInt();
        a = new int[n];
        //Get elements of array
        for(int i = 0; i < a.length; i++)
            { System.out.print("Enter number : ");
            a[i] = sc.nextInt(); }
```

3140705 – Object Oriented Programming -I

```
        }
    //Bubble Sort
    for(int i = 1; i <= a.length; i++)
        { for(int j = 0; j < a.length-i;
        j++) {
            if(a[j]>a[j+1])
                { temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
                }
            }
        }
    //Displaying Sorted Array
    System.out.println("\nSorted array:-");
    for(int i = 0; i < a.length; i++){
        System.out.println(a[i]);
    }
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac BubbleSort.java

E:\darshan college\Java 2019-20\solution\javaprograms>java BubbleSort
Enter size of array : 4
Enter number : 43
Enter number : 32
Enter number : 67
Enter number : 56

Sorted array:-
32
43
56
67

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

5. WAP to read values in twodimensional array and print them in matrix form.

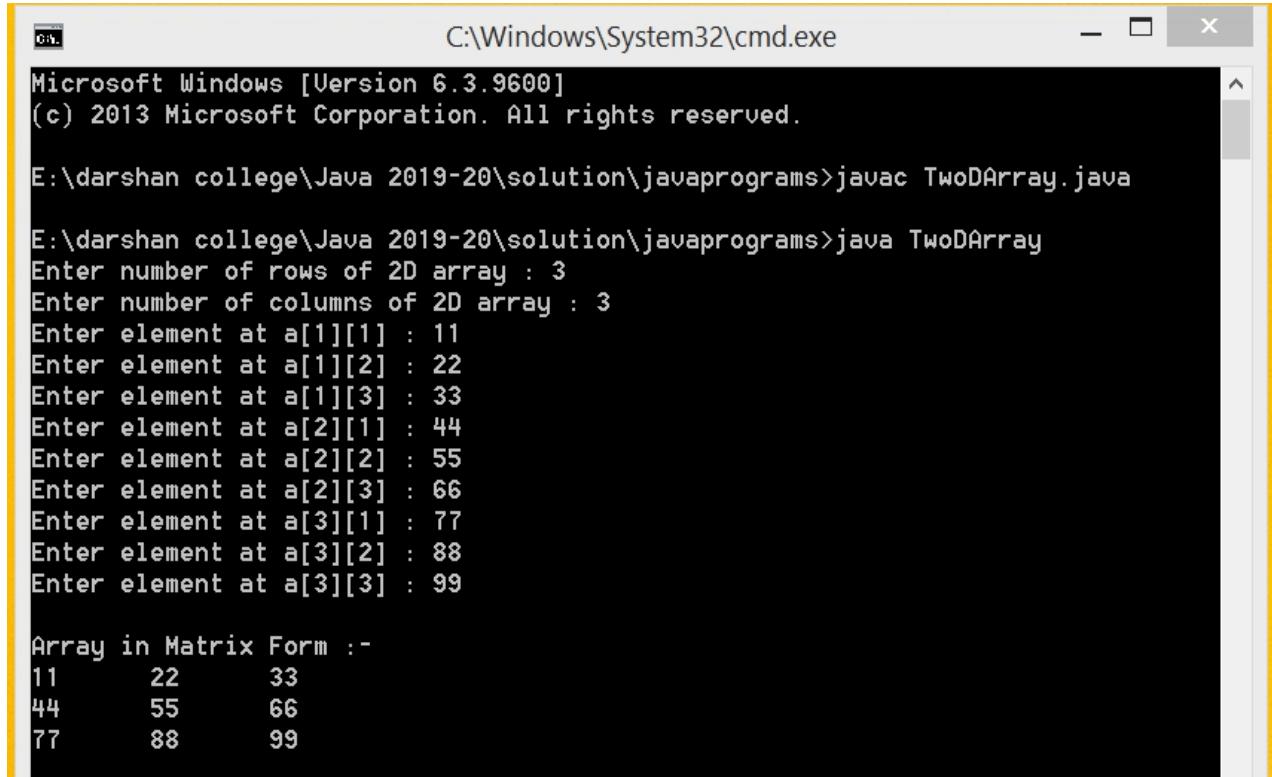
```
//2D array
import java.util.*;

class TwoDArray{
    public static void main(String[] args)
        {int a[][][], row, column;
        Scanner sc = new Scanner(System.in);
        //Get row and column
        System.out.print("Enter number of rows of 2D array : ");
        row = sc.nextInt();
        System.out.print("Enter number of columns of 2D array : ");
        column = sc.nextInt();
        a = new int[row][column];
        //Getting elements of 2D array
        for(int i = 0; i < row; i++){
            for(int j = 0; j < column;
                j++){ System.out.print("Enter element at
                    a["++(i+1)+"]["+(j+1)+"] : ");
                    a[i][j] = sc.nextInt();
                }
        }

        //Displaying 2D array
        System.out.println("\nArray in Matrix Form :-");
        for(int i = 0; i < row; i++){
            for(int j = 0; j < column;
                j++){ System.out.print(a[i][j]+\t");
            }
            System.out.println();
        }
    }
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The output displays the execution of a Java program named 'TwoDArray.java'. The program prompts the user for the number of rows and columns of a 2D array, which is 3 in this case. It then asks for 9 elements to fill the 3x3 matrix. Finally, it prints the matrix in a tabular form.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac TwoDArray.java

E:\darshan college\Java 2019-20\solution\javaprograms>java TwoDArray
Enter number of rows of 2D array : 3
Enter number of columns of 2D array : 3
Enter element at a[1][1] : 11
Enter element at a[1][2] : 22
Enter element at a[1][3] : 33
Enter element at a[2][1] : 44
Enter element at a[2][2] : 55
Enter element at a[2][3] : 66
Enter element at a[3][1] : 77
Enter element at a[3][2] : 88
Enter element at a[3][3] : 99

Array in Matrix Form :-
11      22      33
44      55      66
77      88      99
```

6. WAP to read two matrices of size $n \times n$, perform multiplication operation and store result in third matrix and print it.

```
//Matrix multiplication
import java.util.*;

class MatrixMultiplication{
    public static void main(String[] args)
    { int a[][][], b[][][], c[][][], row,
    column; Scanner sc = new
    Scanner(System.in);
    //Get row and column
    System.out.print("Enter number of rows of 2D array : ");
    row = sc.nextInt();
    System.out.print("Enter number of columns of 2D array : ");
    column = sc.nextInt();

    a = new int[row][column];
    b = new int[row][column];
    c = new int[row][column];
    //Getting elements of 1st matrix
```

3140705 – Object Oriented Programming -I

```
for(int i = 0; i < row; i++){
    for(int j = 0; j < column;
        j++){ System.out.print("Enter element at
        a["++(i+1)+"["++(j+1)+"] : ");
        a[i][j] = sc.nextInt();
    }
}

//Getting elements of 2nd matrix
for(int i = 0; i < row; i++){
    for(int j = 0; j < column;
        j++){ System.out.print("Enter element at
        b["++(i+1)+"["++(j+1)+"] : ");
        b[i][j] = sc.nextInt();
    }
}

//Multiplication
for(int k = 0; k < row;
    k++){ for(int i = 0; i < row;
    i++) {
        c[k][i] = 0;
        for(int j = 0; j < column; j++) {

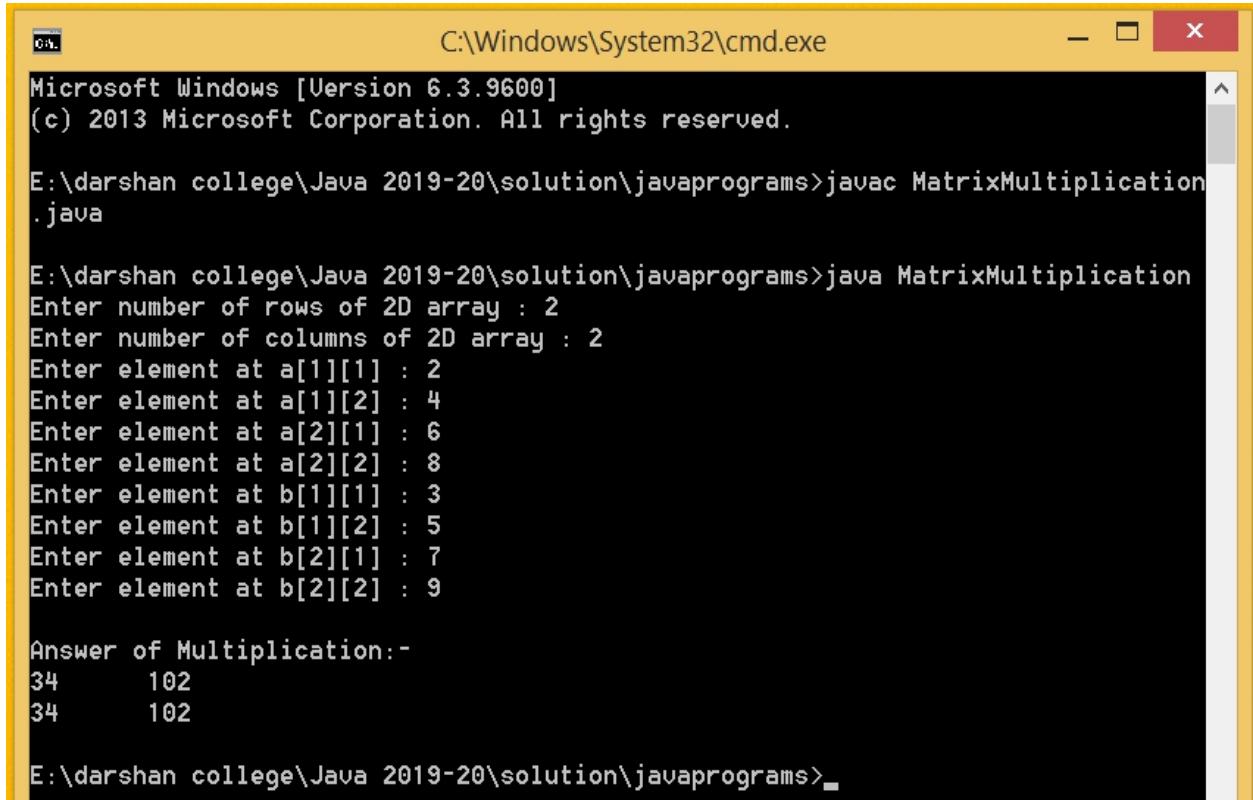
            c[k][i] = c[k][i] + (a[i][j] * b[j][i]);
        }
    }
}

//Displaying answer of multiplication
System.out.println("\nAnswer of Multiplication:-");
for(int i = 0; i < row; i++){
    for(int j = 0; j < column;
        j++){ System.out.print(c[i][j]+\t");
    }
    System.out.println();
}

}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The command 'javac MatrixMultiplication.java' is run from the directory 'E:\darshan college\Java 2019-20\solution\javaprograms'. The program prompts for the dimensions and elements of two 2x2 matrices. The output shows the input for both matrices and the resulting 2x2 matrix product.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac MatrixMultiplication.java

E:\darshan college\Java 2019-20\solution\javaprograms>java MatrixMultiplication
Enter number of rows of 2D array : 2
Enter number of columns of 2D array : 2
Enter element at a[1][1] : 2
Enter element at a[1][2] : 4
Enter element at a[2][1] : 6
Enter element at a[2][2] : 8
Enter element at b[1][1] : 3
Enter element at b[1][2] : 5
Enter element at b[2][1] : 7
Enter element at b[2][2] : 9

Answer of Multiplication:-
34      102
34      102

E:\darshan college\Java 2019-20\solution\javaprograms>
```

7. WAP to accept n numbers in an array. Pass this array to function name selection_sor (int m []). Arrange all the numbers in ascending order using selection sort and display th

```
import java.util.*;
class SelectionSort{
    public static void main(String[] args)
        {int a[], n, temp;

        Scanner sc = new Scanner(System.in);

        //Get size of array
        System.out.print("Enter size of array : ");
        n = sc.nextInt();
        a = new int[n];

        //Get elements of array
        for(int i = 0; i < a.length;
            i++){ System.out.print("Enter
        number : ");a[i] = sc.nextInt();
    }

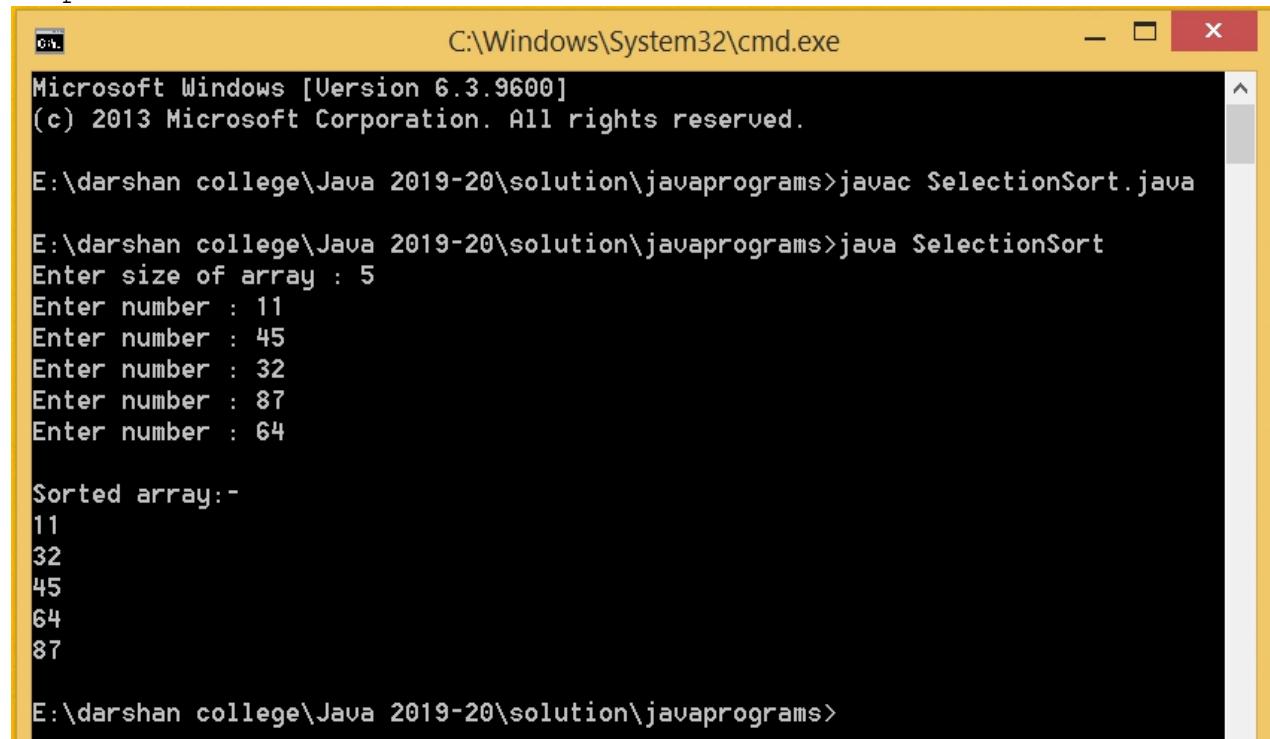
    for(int i = 0; i < n-1;
        i++)
    {
        for(int j = i+1; j < n;
            j++)
        {
            if(a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    for(int i = 0; i < n;
        i++)
    {
        System.out.print(a[i] + " ");
    }
}
```

3140705 – Object Oriented Programming -I

```
//Selection Sort
for(int i = 0; i < a.length;
    i++){ for(int j = i+1; j < a.length;
    j++) {
        if(a[i]>a[j]){
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}

//Displaying Sorted Array
System.out.println("\nSorted array:-");
for(int i = 0; i < a.length; i++){
    System.out.println(a[i]);
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac SelectionSort.java

E:\darshan college\Java 2019-20\solution\javaprograms>java SelectionSort
Enter size of array : 5
Enter number : 11
Enter number : 45
Enter number : 32
Enter number : 87
Enter number : 64

Sorted array:-
11
32
45
64
87

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

8. WAP to store numbers in 4 X 4 matrix in a two-dimensional array. Find the sum of the numbers of each row and the sum of numbers of each column of the matrix.

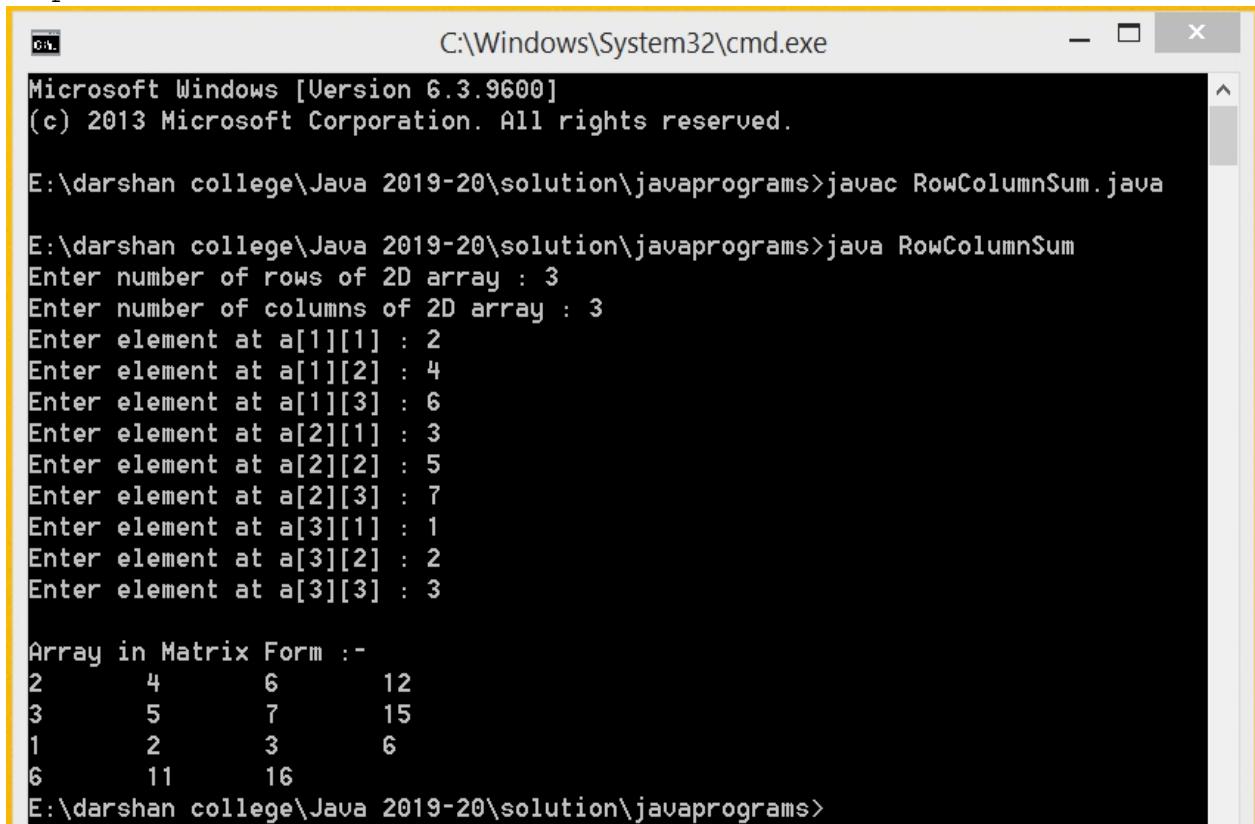
```
import java.util.*;  
  
class RowColumnSum{  
    public static void main(String[] args) {  
        int a[][], row, column, columnSum[], rowSum[];  
        Scanner sc = new Scanner(System.in);  
  
        //Get row and column  
        System.out.print("Enter number of rows of 2D array : ");  
        row = sc.nextInt();  
        System.out.print("Enter number of columns of 2D array : ");  
        column = sc.nextInt();  
  
        a = new int[row][column];  
        rowSum = new int[row];  
        columnSum = new int[column];  
  
        //Getting elements of 2D array  
        for(int i = 0; i < row; i++){  
            for(int j = 0; j < column;  
                j++){ System.out.print("Enter element at  
                a["+ (i+1) +"] [" + (j+1) +"] : ");  
                a[i][j] = sc.nextInt();  
            }  
        }  
  
        //Adding Sum  
        for(int i = 0; i < row;  
            i++){ rowSum[i] = columnSum[i] =  
            0; for(int j = 0; j < column;  
            j++){  
                rowSum[i] += a[i][j];  
                columnSum[i] += a[j][i];  
            }  
        }  
  
        //Displaying 2D array with sums  
        System.out.println("\nArray in Matrix Form :-");  
        for(int i = 0; i <= row ; i++){  
            for(int j = 0; j < column;  
                j++){if(i == row)  
                    System.out.print(columnSum[j] + "\t");  
            }  
        }  
    }  
}
```

3140705 – Object Oriented Programming -I

```
        else
            System.out.print(a[i][j]+"\t");
    }

    if(i != row)
        System.out.println(rowSum[i]);
}
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac RowColumnSum.java

E:\darshan college\Java 2019-20\solution\javaprograms>java RowColumnSum
Enter number of rows of 2D array : 3
Enter number of columns of 2D array : 3
Enter element at a[1][1] : 2
Enter element at a[1][2] : 4
Enter element at a[1][3] : 6
Enter element at a[2][1] : 3
Enter element at a[2][2] : 5
Enter element at a[2][3] : 7
Enter element at a[3][1] : 1
Enter element at a[3][2] : 2
Enter element at a[3][3] : 3

Array in Matrix Form :-
2      4      6      12
3      5      7      15
1      2      3      6
6      11     16

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I**Practical-7**

1. Create a class named Candidate with Candidate_ ID, Candidate_ Name, Candidate_ A Candidate_Weight and Candidate_ Height data members. Also create a method GetCandidateDetails() and DisplayCandidateDetails(). Create main method to demonstrate the Candidate class.

```
import java.util.Scanner;

class Candidate{
    //Declare Variables
    int CandidateID;
    int CandidateAge;
    int CandidateWeight;
    String CandidateName;
    double CandidateHeight;
    Scanner sc = new Scanner(System.in);

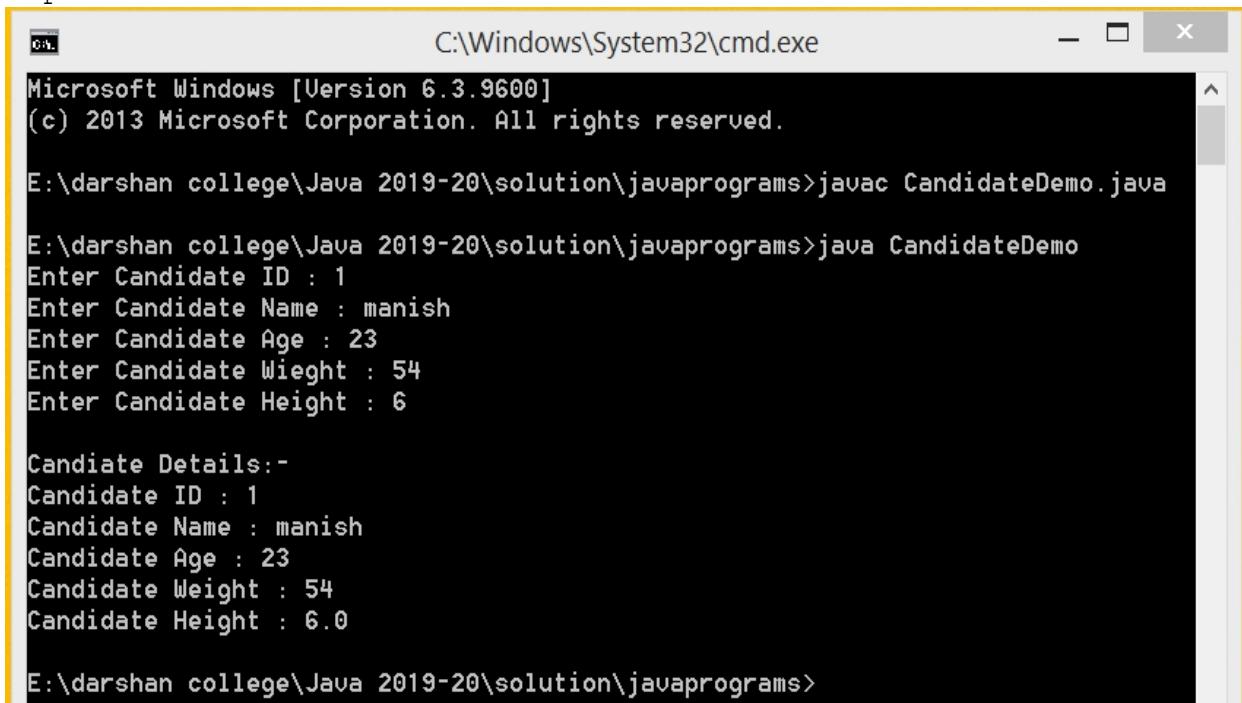
    //Get Details method
    void getCandidateDetails(){ System.out.print("Enter Candidate ID : ");
        CandidateID = sc.nextInt(); System.out.print("Enter Candidate Name : ");
        sc.nextLine(); CandidateName = sc.nextLine();
        System.out.print("Enter Candidate Age : ");
        CandidateAge = sc.nextInt();
        System.out.print("Enter Candidate Weight : ");
        CandidateWeight = sc.nextInt();
        System.out.print("Enter Candidate Height : ");
        CandidateHeight = sc.nextDouble();
    }

    //Display Details method
    void displayCandidateDetails(){ System.out.println("\nCandidate Details:-");
        System.out.println("Candidate ID : "+CandidateID);
        System.out.println("Candidate Name : "+CandidateName);
        System.out.println("Candidate Age : "+CandidateAge);
        System.out.println("Candidate Weight : "+CandidateWeight);
        System.out.println("Candidate Height : "+CandidateHeight);
    }
}
```

3140705 – Object Oriented Programming -I

```
//Main Class
class CandidateDemo{
    //Main method
    public static void main(String[] args) {
        //Object Initialization
        Candidate cn = new Candidate();
        //Call method
        cn.getCandidateDetails();
        cn.displayCandidateDetails();
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac CandidateDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java CandidateDemo
Enter Candidate ID : 1
Enter Candidate Name : manish
Enter Candidate Age : 23
Enter Candidate Wieght : 54
Enter Candidate Height : 6

Candidate Details:-
Candidate ID : 1
Candidate Name : manish
Candidate Age : 23
Candidate Weight : 54
Candidate Height : 6.0

E:\darshan college\Java 2019-20\solution\javaprograms>
```

2. Create a class named `Bank_Account` with `Account_No`, `User_Name`, `Email`, `Account_Type` and `Account_Balance` data members. Also create a method `GetAccountDetails()` and `DisplayAccountDetails()`. Create main method to demonstrate `Bank_Account` class.

```
import java.util.Scanner;

class BankAccount{
    //Declare Variables
    String AccountNumber;
```

3140705 – Object Oriented Programming -I

```
String UserName;
String Email;
String AccountType;
double AccountBalance;
Scanner sc = new Scanner(System.in);

//Get Details method
void
getAccountDetails(){ System.out.print("\nEnter
r Account Number: ");AccountNumber =
sc.next(); System.out.print("Enter User Name :
"); UserName = sc.next();
System.out.print("Enter Email : ");
Email = sc.next();
System.out.print("Enter Account Type : ");
AccountType = sc.next();
System.out.print("Enter Account Balance : ");
AccountBalance = sc.nextDouble();
}

//Display Details method
void
displayAccountDetails(){ System.out.println("\nAccoun
t Details:-"); System.out.println("Account Number:
"+AccountNumber);System.out.println("User Name :
"+UserName); System.out.println("Email : "+Email);
System.out.println("Account Type : "+AccountType);
System.out.println("Account Balance : "+AccountBalance);
}

//Search Method
String searchAccount(String message, String AccountNo,
BankAccount bns){
    //Compare Account Numbers
    if(AccountNo.equals(bns.AccountNumber)){
        message = "Found";
    }
    return message;
}

//Main Class
class BankAccountDemo{
    //Main method
    public static void main(String[] args) {
```

3140705 – Object Oriented Programming -I

```
Scanner sc = new Scanner(System.in);
//Object Initialization
BankAccount bn1 = new BankAccount();
BankAccount bn2 = new BankAccount();
BankAccount bn3 = new BankAccount();
BankAccount bn4 = new BankAccount();
BankAccount bn5 = new BankAccount();

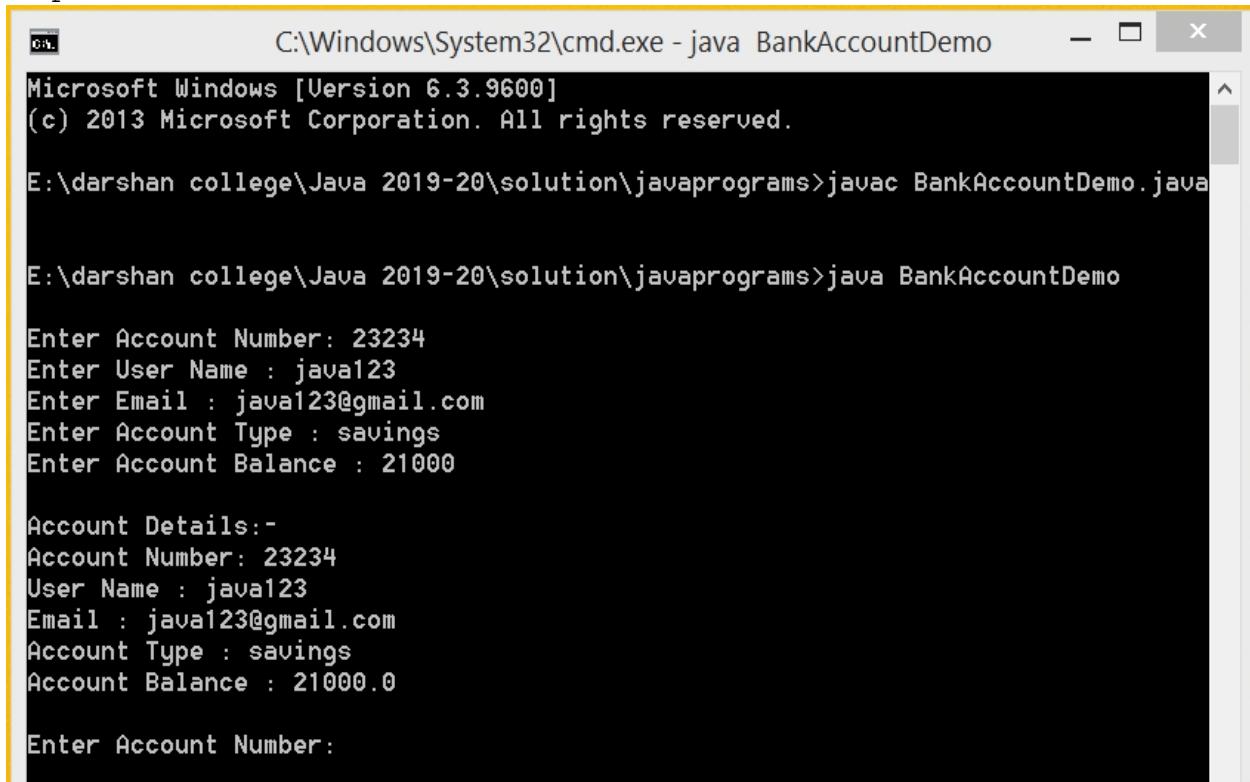
//Call method
bn1.getAccountDetails();
bn1.displayAccountDetails();
bn2.getAccountDetails();
bn2.displayAccountDetails();
bn3.getAccountDetails();
bn3.displayAccountDetails();
bn4.getAccountDetails();
bn4.displayAccountDetails();
bn5.getAccountDetails();
bn5.displayAccountDetails();

String message = "Not Found";
System.out.print("\nEnter Account Number to search: ");
String AccountNo = sc.next();
message = bn1.searchAccount(message, AccountNo, bn1);
message = bn2.searchAccount(message, AccountNo, bn2);
message = bn3.searchAccount(message, AccountNo, bn3);
message = bn4.searchAccount(message, AccountNo, bn4);
message = bn5.searchAccount(message, AccountNo, bn5);

//Displaying Search Results
System.out.println(message);
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe - java BankAccountDemo'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac BankAccountDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java BankAccountDemo

Enter Account Number: 23234
Enter User Name : java123
Enter Email : java123@gmail.com
Enter Account Type : savings
Enter Account Balance : 21000

Account Details:-
Account Number: 23234
User Name : java123
Email : java123@gmail.com
Account Type : savings
Account Balance : 21000.0

Enter Account Number:
```

3. WAP with followings specifications:

Class Name: Employee

Data Members: Employee_ID, Employee_Name, Designation, Age, Salary

Member Functions: GetEmployeeDetails () and DisplayEmployeeDetails ().

```
import java.util.Scanner;

class Employee{
    //Declare Variables
    int EmployeeID;
    int Age;
    int Salary;
    String EmployeeName, Designation;
    Scanner sc = new Scanner(System.in);

    //Get Details method
    void
    getEmployeeDetails(){ System.out.print("Enter Employee ID : ");
        EmployeeID =
        sc.nextInt(); System.out.print("Enter Employee Name : ");
        EmployeeName = sc.nextLine(); }

    //Display Details method
    void
    DisplayEmployeeDetails(){ System.out.print("Employee ID : " +
        EmployeeID + " Employee Name : " + EmployeeName + " Designation : " +
        Designation + " Age : " + Age + " Salary : " + Salary); }
```

3140705 – Object Oriented Programming -I

```
EmployeeName = sc.next();
System.out.print("Enter Designation : ");
Designation = sc.next();
System.out.print("Enter Age : ");
Age = sc.nextInt();
System.out.print("Enter Salary : ");
Salary = sc.nextInt();
}

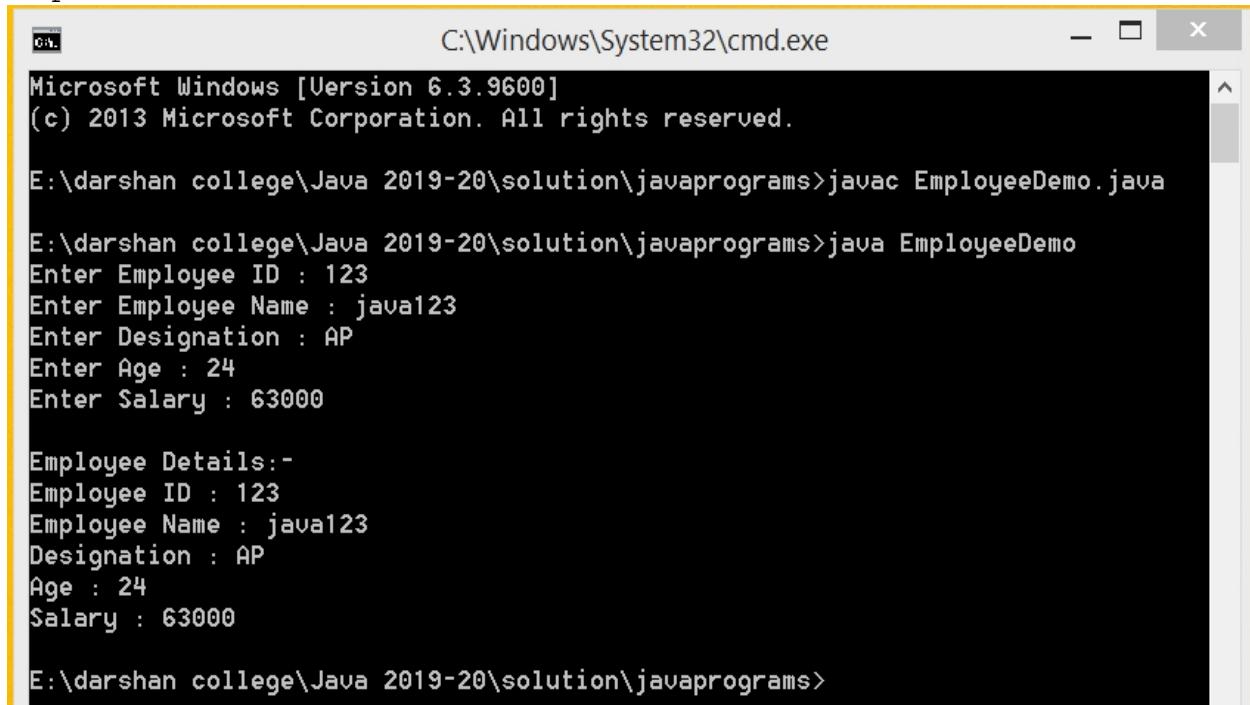
//Display Details method
void
displayEmployeeDetails() { System.out.println("\nEmployee Details:-");
System.out.println("Employee ID : "+EmployeeID);
System.out.println("Employee Name : "+EmployeeName);
System.out.println("Designation : "+Designation);
System.out.println("Age : "+Age);
System.out.println("Salary : "+Salary);
}

}

//Main Class
class EmployeeDemo{
    //Main method
    public static void main(String[] args) {
        //Object Initialization
        Employee e = new Employee();
        //Call method
        e.getEmployeeDetails();
        e.displayEmployeeDetails();
    }
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac EmployeeDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java EmployeeDemo
Enter Employee ID : 123
Enter Employee Name : java123
Enter Designation : AP
Enter Age : 24
Enter Salary : 63000

Employee Details:-
Employee ID : 123
Employee Name : java123
Designation : AP
Age : 24
Salary : 63000

E:\darshan college\Java 2019-20\solution\javaprograms>
```

4. Write a class program with following specifications:

Class Name: Student

Data Members: Enrollment_No, Student_Name, Semester, CPI and SPI

Member Functions: GetStudentDetails () and DisplayStudentDetails ().

```
import java.util.Scanner;

class Student{
    //Declare Variables
    int EnrollmentNo;
    int Semester;
    String Name;
    double CPI;
    double SPI;
    Scanner sc = new Scanner(System.in);

    //Get Details method
    void
        getStudentDetails() { System.out.print("Enter Enrollment No : "); EnrollmentNo =
        sc.nextInt(); System.out.print("Enter Name : ");
        Name = sc.next(); }
```

3140705 – Object Oriented Programming -I

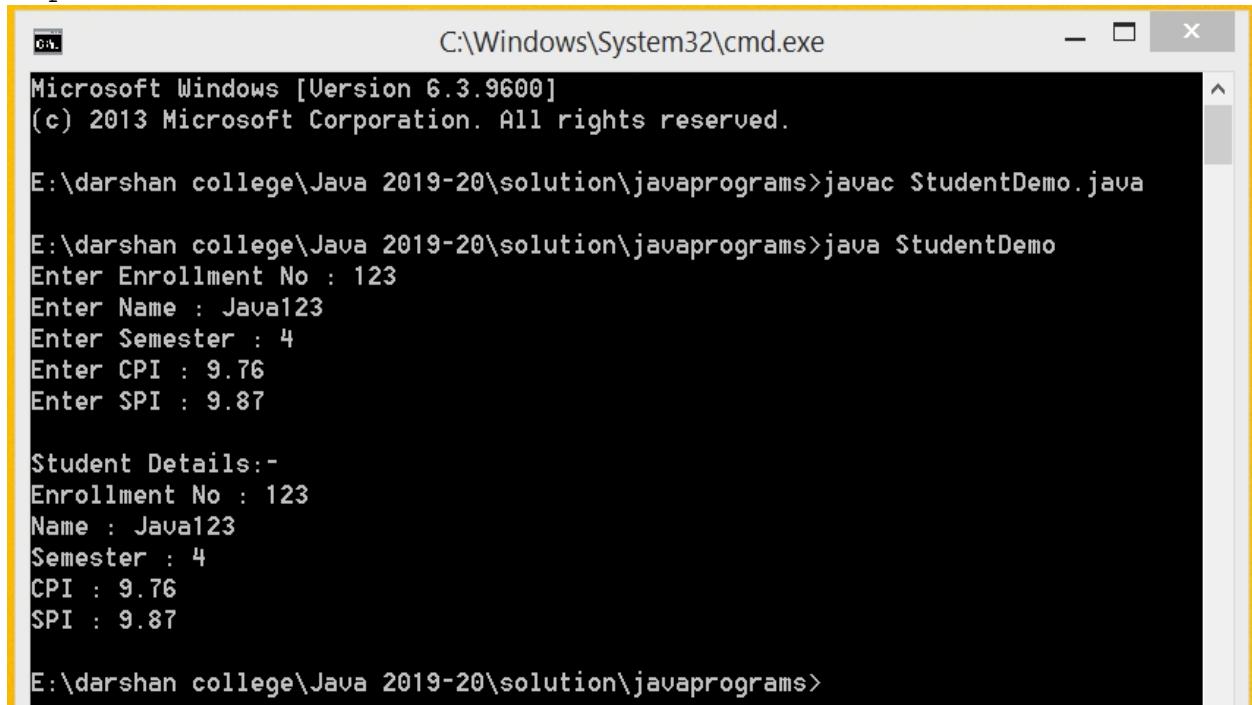
```
System.out.print("Enter Semester : ");
Semester = sc.nextInt();
System.out.print("Enter CPI : ");
CPI = sc.nextDouble();
System.out.print("Enter SPI : ");
SPI = sc.nextDouble();
}

//Display Details method
void
displayStudentDetails(){ System.out.println("\nStudent Details:-"); System.out.println("Enrollment No : "+EnrollmentNo);System.out.println("Name : "+Name);
System.out.println("Semester : "+Semester);
System.out.println("CPI : "+CPI);
System.out.println("SPI : "+SPI);
}
}

//Main Class
class StudentDemo{
    //Main method
    public static void main(String[] args) {
        //Object Initialization
        Student st = new Student();
        //Call method
        st.getStudentDetails();
        st.displayStudentDetails();
    }
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac StudentDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java StudentDemo
Enter Enrollment No : 123
Enter Name : Java123
Enter Semester : 4
Enter CPI : 9.76
Enter SPI : 9.87

Student Details:-
Enrollment No : 123
Name : Java123
Semester : 4
CPI : 9.76
SPI : 9.87

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-8

1. WAP to create Circle class with area and perimeter function to find area and perimete circle.

```
import java.util.Scanner;

// Class Circle
class Circle{
    //Declare variables
    double radius;
    double areaOfCircle;
    double perimiterOfCircle;
    final double pi = 3.14;
    Scanner sc = new Scanner(System.in);

    //Get Radius Method
    void getRadius(){
        System.out.print("Enter Radius : ");
        radius = sc.nextDouble();
    }

    //Find Area Method
    void areaOfCircle(){
        areaOfCircle = pi * radius * radius;
    }

    //Find Perimiter Method
    void perimiterOfCircle(){
        perimiterOfCircle = 2 * pi * radius;
    }

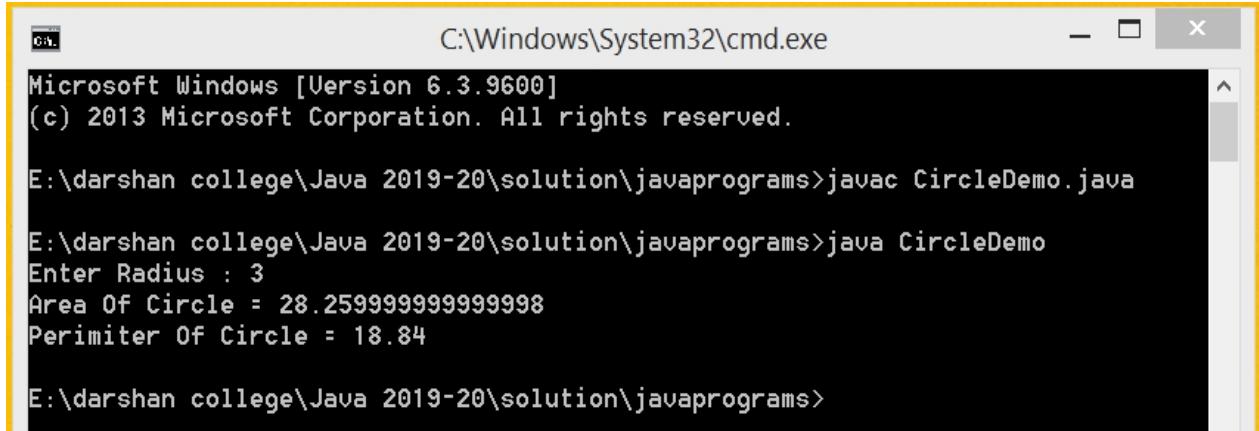
    //Display Results
    void display(){
        System.out.println("Area Of Circle = "+areaOfCircle);
        System.out.println("Perimiter      Of      Circle      = "      +
perimiterOfCircle);
    }
}

//Main Class
class CircleDemo{
    //Main Method
    public static void main(String[] args) {
```

3140705 – Object Oriented Programming -I

```
//Object Initialization  
Circle c = new Circle();  
  
//Call Method  
c.getRadius();  
c.areaOfCircle();  
c.perimiterOfCircle();  
c.display();  
}  
}
```

Output:



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
  
E:\darshan college\Java 2019-20\solution\javaprograms>javac CircleDemo.java  
  
E:\darshan college\Java 2019-20\solution\javaprograms>java CircleDemo  
Enter Radius : 3  
Area Of Circle = 28.25999999999998  
Perimiter Of Circle = 18.84  
  
E:\darshan college\Java 2019-20\solution\javaprograms>
```

2. Define Time class with hour and minute as data member. Also define addition method to add two time objects.

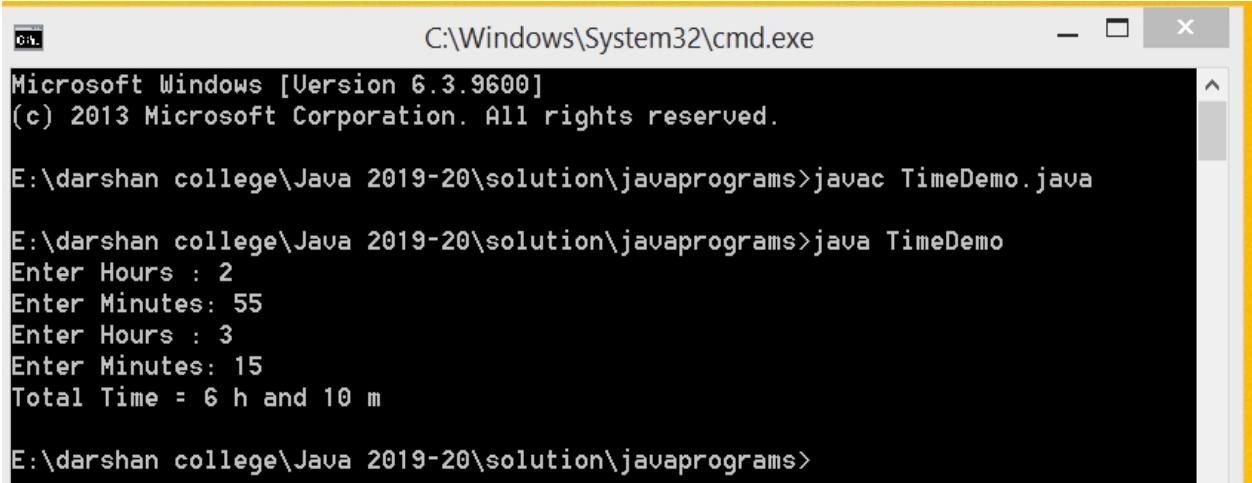
```
import java.util.Scanner;  
  
// Class Time  
class Time{  
    //Declare variables  
    int hours;  
    int minutes;  
    Scanner sc = new Scanner(System.in);  
  
    //Get Details Method  
    void getDetails(){  
        do{  
            System.out.print("Enter Hours : ");  
            hours = sc.nextInt();  
            System.out.print("Enter Minutes: ");  
            minutes = sc.nextInt();  
        } while(hours < 0 || hours > 23 || minutes < 0 || minutes > 59);  
    }  
  
    //Set Details Method  
    void setDetails(int h, int m){  
        hours = h;  
        minutes = m;  
    }  
  
    //Get Hours Method  
    int getHours(){  
        return hours;  
    }  
  
    //Get Minutes Method  
    int getMinutes(){  
        return minutes;  
    }  
  
    //Add Two Time Objects  
    Time addTime(Time t2){  
        int totalHours = hours + t2.hours;  
        int totalMinutes = minutes + t2.minutes;  
        if(totalMinutes > 59){  
            totalHours++;  
            totalMinutes -= 60;  
        }  
        Time t3 = new Time();  
        t3.hours = totalHours;  
        t3.minutes = totalMinutes;  
        return t3;  
    }  
}
```

3140705 – Object Oriented Programming -I

```
        }while(minutes >= 60);  
    }  
  
    //Add Time objects Method  
    Time addTime(Time t2){  
        //Object Initialization  
        Time t3 = new Time();  
        //Add Logic  
        t3.hours = hours + t2.hours;  
        t3.minutes = minutes + t2.minutes;  
        if(t3.minutes >= 60){  
            t3.hours++;  
            t3.minutes -= 60;  
        }  
        return t3;  
    }  
  
    //Display Results  
    void display(){  
        System.out.println("Total Time = "+hours+" h and " +  
        minutes + " m");  
    }  
}  
  
//Main Class  
class TimeDemo{  
    //Main Method  
    public static void main(String[] args) {  
        //Object Intiatlization  
        Time t1 = new Time();  
        Time t2 = new Time();  
        Time t3;  
        //Call Method  
        t1.getDetails();  
        t2.getDetails();  
        t3 = t1.addTime(t2);  
        t3.display();  
    }  
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac TimeDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java TimeDemo
Enter Hours : 2
Enter Minutes: 55
Enter Hours : 3
Enter Minutes: 15
Total Time = 6 h and 10 m

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3. Define class for Complex number with real and imaginary part. Describe its construct overload the constructors and instantiate its object. Also define addition method to add Complex objects.

```
import java.util.Scanner;

// Class ComplexNumbers
class ComplexNumbers{
    //Declare variables
    int real;
    int imaginary;
    int totalReal, totalImaginary;
    Scanner sc = new Scanner(System.in);

    //Constructor
    ComplexNumbers(int i){
        System.out.println("Enter Complex Numbers "+i);
        System.out.print("Enter Real : ");
        real = sc.nextInt();
        System.out.print("Enter Imaginary : ");
        imaginary = sc.nextInt();
    }

    //Add Time objects Method
    void addComplexNumbers(ComplexNumbers cn2){
        //Add Logic
        totalReal = real + cn2.real;
        totalImaginary = imaginary + cn2.imaginary;
    }
}
```

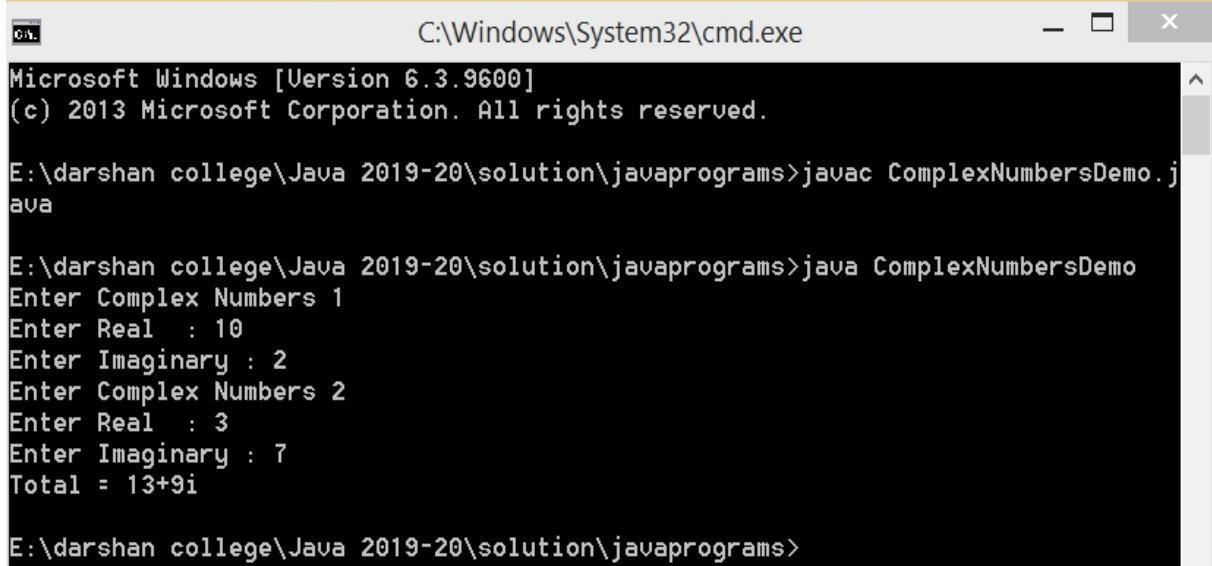
3140705 – Object Oriented Programming -I

```
}

//Display Results
void display(){
    System.out.println("Total      =      "+      totalReal      +      "+"      +
totalImaginary + "i");
}
}

//Main Class
class ComplexNumbersDemo{
    //Main Method
    public static void main(String[] args) {
        //Object Initialization
        ComplexNumbers cn1 = new ComplexNumbers(1);
        ComplexNumbers cn2 = new ComplexNumbers(2);
        //Call Method
        cn1.addComplexNumbers(cn2);
        cn1.display();
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac ComplexNumbersDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java ComplexNumbersDemo
Enter Complex Numbers 1
Enter Real : 1
Enter Imaginary : 2
Enter Complex Numbers 2
Enter Real : 3
Enter Imaginary : 7
Total = 13+9i

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-9

1. Create first class with following specifications:

Class Name: Student

Data Members: Enrollment_No, Student_Name, Semester

Member Functions: GetStudentDetails ()

Create second class with following specifications:

Class Name: Result

Data Members: Enrollment_No, CPI and SPI

Member Functions: GetResultDetails () and DisplayResult ()

DisplayResult method header must be: public void DisplayResult (Student s).

```
import java.util.*;  
  
//Class Student  
class Student{  
    //Declare Variables  
    int enrollmentNo;  
    String studentName;  
    String course;  
    Scanner sc = new Scanner(System.in);  
    //Method get student details  
    void getStudentDetails(){  
        System.out.println("Student Details :-");  
        System.out.print("Enter Enrollment No : ");  
        enrollmentNo = sc.nextInt();  
        System.out.print("Enter Student Name : ");  
        studentName = sc.next();  
        System.out.print("Enter Course : ");  
        course = sc.next();  
    }  
}  
  
//Class Result  
class Result{  
    //Declare Variables  
    int enrollmentNo;  
    int sem;  
    double CPI;  
    double SPI;  
    Scanner sc = new Scanner(System.in);  
}
```

3140705 – Object Oriented Programming -I

```
//Method Get Result Details
void getResultDetails(){
    System.out.println("Result Details :-");
    System.out.print("Enter Enrollment No : ");
    enrollmentNo = sc.nextInt();
    System.out.print("Enter Semester : ");
    sem = sc.nextInt();
    System.out.print("Enter SPI : ");
    SPI = sc.nextDouble();
    System.out.print("Enter CPI : ");
    CPI = sc.nextDouble();
}

//Method Display Result Details
void displayResultDetails(Student st[], int
    i){ System.out.println("Enrollment No :
"+enrollmentNo);System.out.println("Name :
"+st[i].studentName); System.out.println("Course :
"+st[i].course); System.out.println("Semester :
"+sem); System.out.println("SPI : "+SPI);
    System.out.println("CPI : "+CPI);
}

//Main Class
class ResultDemo{
    //Main Method
    public static void main(String[] args) {
        //Declare Variables
        int n;
        int enrollmentNumber;
        int i;
        boolean found = false;
        Scanner sc = new Scanner(System.in);
        //Get number of students
        System.out.print("Enter number of students : ");
        n = sc.nextInt();
        //Object Initialization
        Student st[] = new Student[n];
        Result rs[] = new Result[n];
        for(i = 0; i < n; i++){
            st[i] = new Student();
            rs[i] = new Result();
            st[i].getStudentDetails();
            rs[i].getResultDetails();
```

3140705 – Object Oriented Programming -I

```
}

//Find Student if exist
System.out.print("Enter Enrollment No to get details : ");
enrollmentNumber = sc.nextInt();
for(i = 0; i < n; i++){
    if(enrollmentNumber ==
        st[i].enrollmentNo){found = true;
        break;
    }
}

//Display Result
if(found == true){
    rs[i].displayResultDetails(st, i);
}
else{
    System.out.print("Not Found!");
}
}
```

Output:

```
C:\Windows\System32\cmd.exe
Enter number of students : 2
Student Details :-
Enter Enrollment No : 123
Enter Student Name : java123
Enter Course : bece
Result Details :-
Enter Enrollment No : 123
Enter Semester : 4
Enter SPI : 9.8
Enter CPI : 8.7
Student Details :-
Enter Enrollment No : 456
Enter Student Name : java456
Enter Course : bece
Result Details :-
Enter Enrollment No : 456
Enter Semester : 4
Enter SPI : 9.8
Enter CPI : 9.7
Enter Enrollment No to get details : 123
Enrollment No : 123
Name : java123
Course : bece
Semester : 4
SPI : 9.8
```

3140705 – Object Oriented Programming -I

2. Create array of object for Student_Detail (Enrollment_no, Name, Sem, CPI) class for students, read their information and print it.

```
import java.util.Scanner;

class Student_Detail{
    //Declare Variables
    int EnrollmentNo;
    int Semester;
    String Name;
    double CPI;
    Scanner sc = new Scanner(System.in);

    //Get Details method
    void
        getStudentDetails(){ System.out.print("Enter Enrollment No : ");EnrollmentNo =
        sc.nextInt(); System.out.print("Enter Name : ");
        Name = sc.next();
        System.out.print("Enter Semester : ");
        Semester = sc.nextInt();
        System.out.print("Enter CPI : ");
        CPI = sc.nextDouble();
    }

    //Display Details method
    void
        displayStudentDetails(){ System.out.println("Enrollment No : "+EnrollmentNo);System.out.println("Name : "+Name); System.out.println("Semester : "+Semester);
        System.out.println("CPI : "+CPI);
    }
}

//Main Class
class ArrayStudent{
    //Main method
    public static void main(String[] args) {
        //Declare Variables
        int i, numberOfStudents;
        Scanner sc = new Scanner(System.in);
        //Get number of students
        System.out.print("Enter number of students : ");
        numberOfStudents = sc.nextInt();
    }
}
```

3140705 – Object Oriented Programming -I

```
//Object Initialization
Student_Detail st[] = new Student_Detail[numberOfStudents];

//Get students details
for(i = 0; i < numberOfStudents;
    i++){st[i] = new Student_Detail();
    System.out.println("\nEnter student-"+ (i+1) + " details");
    st[i].getStudentDetails();
}

//Print students details
for(i = 0; i < numberOfStudents;
    i++){ System.out.println("\nStudent-"+ (i+1) + " details");st[i].displayStudentDetails();
}

}
```

Output:

```
C:\Windows\System32\cmd.exe
Enter number of students : 2

Enter student-1 details
Enter Enrollment No : 1
Enter Name : xyz
Enter Semester : 4
Enter CPI : 9.6

Enter student-2 details
Enter Enrollment No : 2
Enter Name : pqr
Enter Semester : 4
Enter CPI : 9.8

Student-1 details
Enrollment No : 1
Name : xyz
Semester : 4
CPI : 9.6

Student-2 details
Enrollment No : 2
Name : pqr
Semester : 4
CPI : 9.8
```

3140705 – Object Oriented Programming -I

3. WAP that counts the number of objects created by using static variable.

```
import java.util.Scanner;

class CountObject{
    //Declare Variables
    static int noOfObjects = 0;

    // Instead of performing increment in the constructor
    // instance block is preferred to make this program generic.
    {
        noOfObjects += 1;
    }

    public static void printObjectCreated()
    {
        System.out.println("Number      of      object      created:      "      +
                           noOfObjects);
    }
}

//Main Class
class CountObjectDemo{

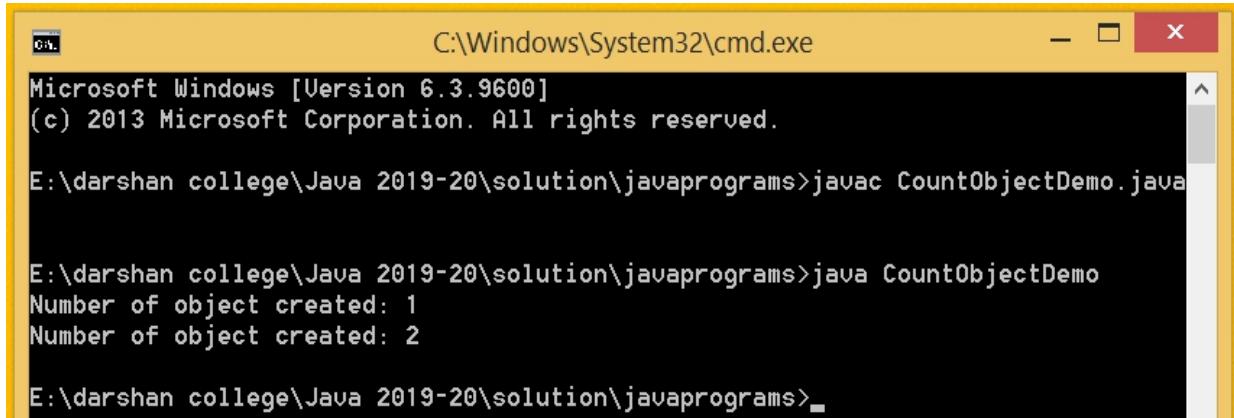
    //Main method
    public static void main(String[] args) {
        //Object Initialization
        CountObject o1 = new CountObject();

        //Call method to print number of objects created
        CountObject.printObjectCreated();

        CountObject o2 = new CountObject();
        CountObject.printObjectCreated();
    }
}
```

3140705 – Object Oriented Programming -I

Output:



A screenshot of a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window shows the following text output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac CountObjectDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java CountObjectDemo
Number of object created: 1
Number of object created: 2

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I**Practical-10**

1. Create a class named 'Member' having the following members:

- 1 – Name
- 2 – Age
- 3 – Phone number
- 4 – Address
- 5 – Salary

It also has a method named 'printSalary' which prints the salary of the members.

Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager making an object of both of these classes and print the same along with specialization department respectively.

```
import java.util.Scanner;

//Class Member
class Member{
    //Declare variables
    String name;
    int age;
    int phoneNumber;
    String address;
    double salary;

    //printSalary Method
    void printSalary(){
        System.out.println("salary = "+salary);
    }
}

//Class Employee
class Employee extends Member{
    //Declare variables
    String specialization;

    //Parameterized Constructor
    Employee(String n, int a, int ph, String add, double s, String spe){
        name=n;
```

3140705 – Object Oriented Programming -I

```
age=a;
phoneNumber=ph;
address=add;
salary=s;
specialization=spe;
}

//Method for display employee details
void displayEmployeeDetails() {
    System.out.println("\nEmployee Details:");
    System.out.println("Name = "+name);
    System.out.println("Age = "+age);
    System.out.println("Phone Number = "+phoneNumber);
    System.out.println("Address = "+address);
    printSalary();
    System.out.println("Specialization = "+specialization);
}
}

//Class Manager
class Manager extends Member{
    //Declare variables
    String department;

    //Parameterized Contructor
    Manager(String n, int a, int ph, String add, double s, String dept) {
        name=n;
        age=a;
        phoneNumber=ph;
        address=add;
        salary=s;
        department=dept;
    }

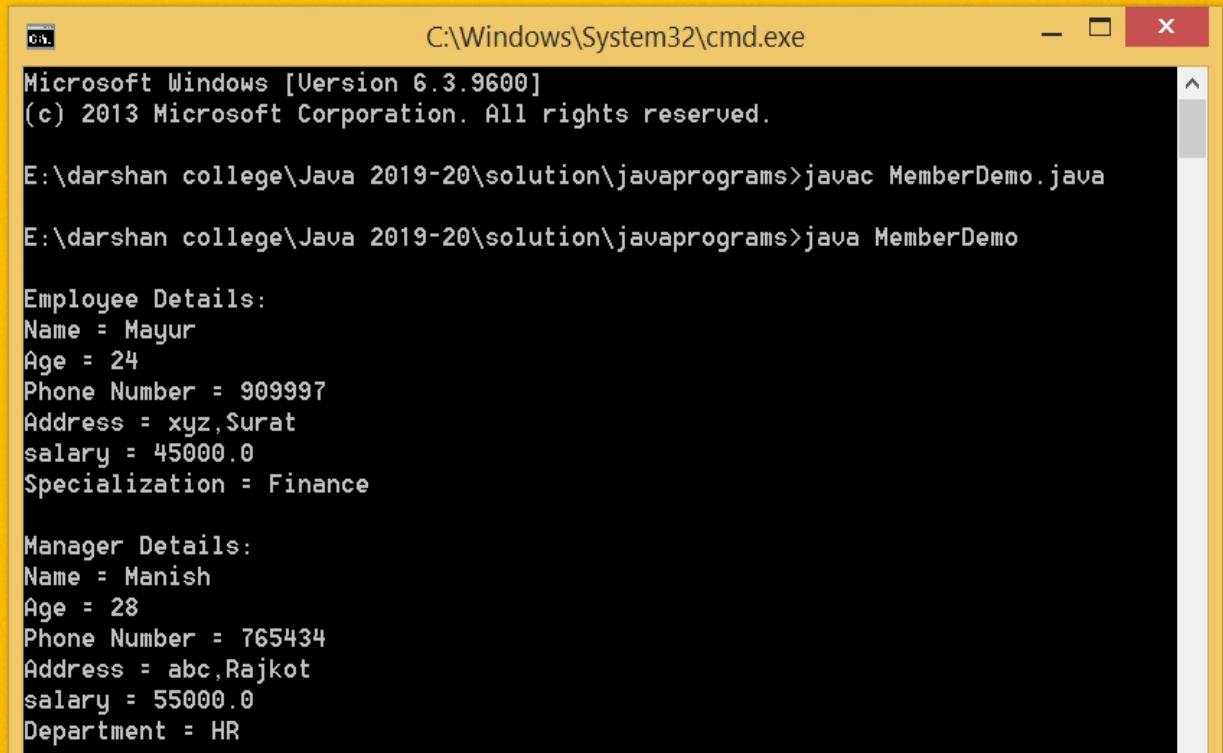
    //Method for display manager details
    void displayManagerDetails(){
        System.out.println("\nManager Details:");
        System.out.println("Name = "+name);
        System.out.println("Age = "+age);
        System.out.println("Phone Number = "+phoneNumber);
        System.out.println("Address = "+address);
        printSalary();
        System.out.println("Department = "+department);
    }
}
```

3140705 – Object Oriented Programming -I

```
}

//Main Class
class MemberDemo{
    //Main Method
    public static void main(String[] args) {
        Employee e1 = new Employee("Mayur", 24, 909997, "xyz, Surat",
        45000, "Finance");
        Manager m1 = new Manager("Manish", 28, 765434, "abc, Rajkot",
        55000, "HR");
        e1.displayEmployeeDetails();
        m1.displayManagerDetails();
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac MemberDemo.java
E:\darshan college\Java 2019-20\solution\javaprograms>java MemberDemo

Employee Details:
Name = Mayur
Age = 24
Phone Number = 909997
Address = xyz, Surat
salary = 45000.0
Specialization = Finance

Manager Details:
Name = Manish
Age = 28
Phone Number = 765434
Address = abc, Rajkot
salary = 55000.0
Department = HR
```

3140705 – Object Oriented Programming -I

2. Design a class named MyPoint to represent a point with x and y-coordinates. The class contains:

The data fields x and y that represent the coordinates with getter methods.

- a no-arg constructor that creates a point (0, 0).
- a constructor that constructs a point with specified coordinates.
- a method named distance that returns the distance from this point to a specific point of the MyPoint type.
- a method named distance that returns the distance from this point to another point with specified x- and y-coordinates.

Create a class named ThreeDPoint to model a point in a three-dimensional space. Let ThreeDPoint be derived from MyPoint with following additional features:

- a data field named z that represents the coordinate.
- a no-arg constructor that creates a point (0, 0, 0).
- a constructor that constructs a point with three specified coordinates.
- a get method that returns the z value.
- Override the distance method to return the distance between two points in the three-dimensional space.

Write a program that creates two points (0, 0, 0) and (10, 30, 25.5) and display the distance between the two points.

```
import java.util.Scanner;
import java.lang.Math;

//Class MyPoint
class MyPoint{
    //Declare Variables
    double x;
    double y;
    Scanner sc = new Scanner(System.in);

    //Default Constructor
    MyPoint() {
        x = 0;
        y = 0;
    }

    //Parameterized Constructor
    MyPoint(double x, double y) {
        this.x = x;
        this.y = y;
    }
}
```

3140705 – Object Oriented Programming -I

```
}

//Get co-ordinates method
void getter(){
    System.out.print("X = "+x+"\nY = "+y);
}

//Distance Method
void distance(double x, double y){
    double distance = Math.sqrt((this.x-x)*(this.x-x) +
    (this.y-y)*(this.y-y));
    System.out.println("Distance is "+distance);
}

//Distance Method
void distance(MyPoint mp2){
    double distance = Math.sqrt((x-mp2.x)*(x-mp2.x) + (y-
    mp2.y)*(y-mp2.y));
    System.out.println("Distance is "+distance);
}

//Class ThreeDPoints
class ThreeDPoints extends MyPoint{
    //Declare Variables
    double z;
    //Default Constructor
    ThreeDPoints(){
        super(0,0);
        z = 0;
    }

    //Parameterized Constructor
    ThreeDPoints(double x, double y, double z){
        super(x,y);
        this.z = z;
    }

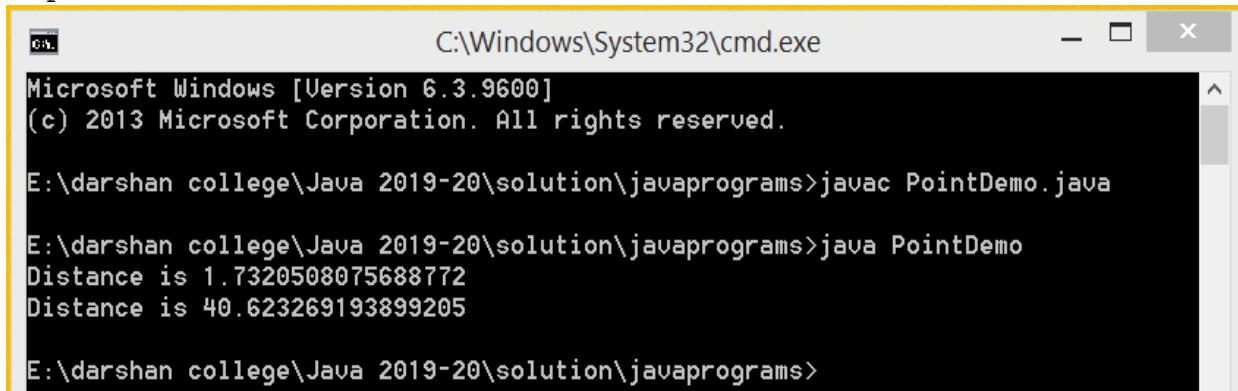
    //Distance Method
    void distance(double x, double y, double z){
        double distance = Math.pow(Math.pow(this.x - x, 2) +
        Math.pow(this.y - y, 2) + Math.pow(this.z - z, 2) * 1.0),
        0.5;
        System.out.println("Distance is "+distance);
    }
}
```

3140705 – Object Oriented Programming -I

```
//Distance Method
void distance(ThreeDPoints tp2) {
    double distance = Math.pow((Math.pow(x - tp2.x, 2) +
    Math.pow(y - tp2.y, 2) + Math.pow(z - tp2.z, 2)) * 1.0),
    0.5);
    System.out.println("Distance is "+distance);
}
}

//Main Class
class PointDemo{
    //Main Method
    public static void main(String[] args)
    { ThreeDPoints tp1 = new ThreeDPoints();
    ThreeDPoints tp2 = new ThreeDPoints(10,30,25.5);
    tp1.distance(1,1,1);
    tp1.distance(tp2);
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac PointDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java PointDemo
Distance is 1.7320508075688772
Distance is 40.623269193899205

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-11

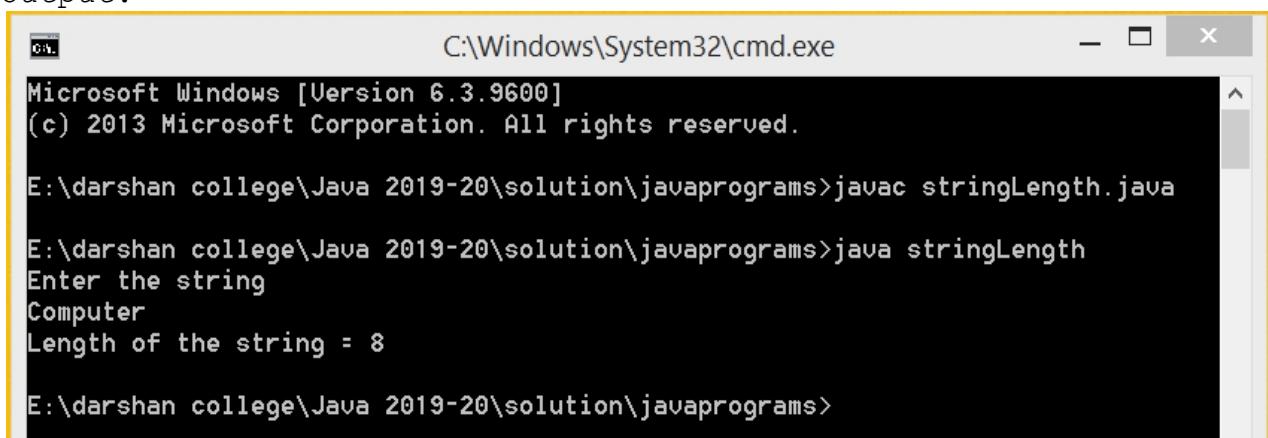
1. WAP to find length of a string without using built function.

```
import java.util.*;
//Main class
class stringLength{
    public static void main(String args[]) {
        //Declare Variables
        int i=0;
        String str;
        Scanner sc = new Scanner(System.in);

        //Get String
        System.out.println("Enter the string");
        str=sc.nextLine();
        char ch[]=str.toCharArray();

        for(char c :
            ch) {i++;
        }
        System.out.println("Length of the string = "+i);
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\System32\cmd.exe'. The window displays the following output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac stringLength.java

E:\darshan college\Java 2019-20\solution\javaprograms>java stringLength
Enter the string
Computer
Length of the string = 8

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

2. WAP that checks whether a given string is a palindrome or not.

```
import java.util.Scanner;

//Main class
class StringPalindrome{
    public static void main(String[ ] arg) {
        //Declare Variables
        String str, rev = "";
        Scanner sc = new Scanner(System.in);

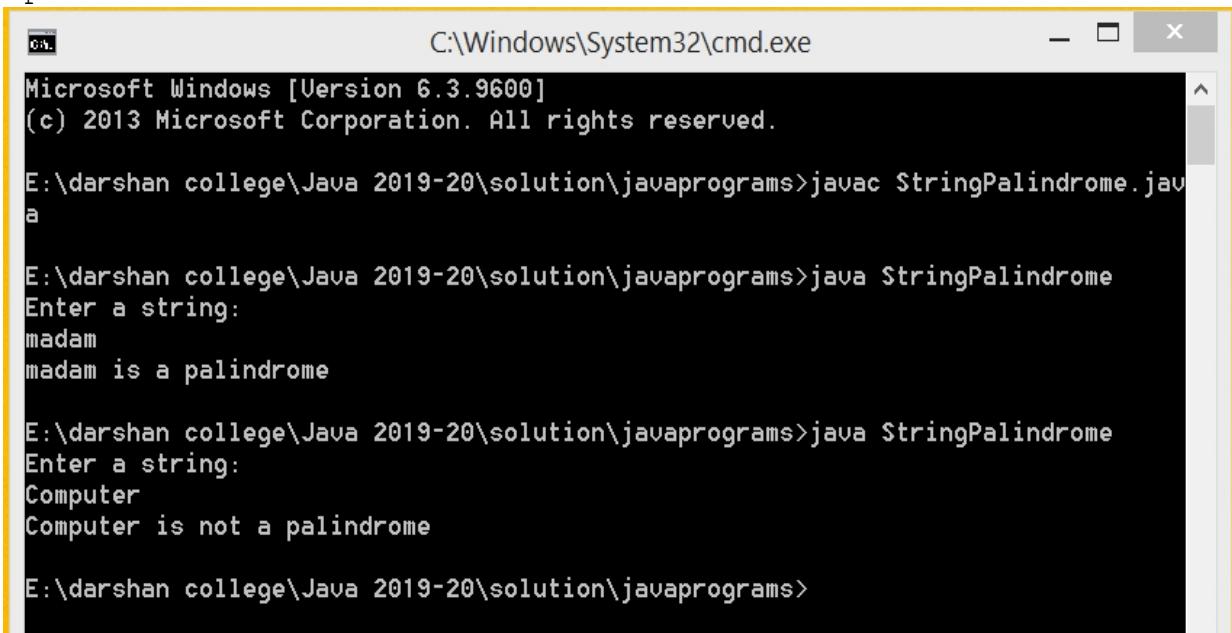
        System.out.println("Enter a string:");
        str = sc.nextLine();

        int length = str.length();

        for ( int i = length - 1; i >= 0; i-- )
            rev = rev + str.charAt(i);

        if (str.equals(rev))
            System.out.println(str+" is a palindrome");
        else
            System.out.println(str+" is not a palindrome");
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac StringPalindrome.java

E:\darshan college\Java 2019-20\solution\javaprograms>java StringPalindrome
Enter a string:
madam
madam is a palindrome

E:\darshan college\Java 2019-20\solution\javaprograms>java StringPalindrome
Enter a string:
Computer
Computer is not a palindrome

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

3. WAP to accept a string and display all the vowels present in the word.

```
import java.util.Scanner;

//Main class
class stringVowels{
    public static void main(String[ ] arg) {
        //Declare Variables
        String s;
        char ch;
        int i=0;
        Scanner sc=new Scanner(System.in);

        //Get String
        System.out.print("Enter a string : ");
        s=sc.nextLine();
        System.out.println("Vowels in a string are:");
        for(int j=0;j<s.length();j++){
            ch=s.charAt(j);
            switch(ch){
                case 'a' :
                case 'e' :
                case 'i' :
                case 'o' :
                case 'u' :
                case 'A' :
                case 'E' :
                case 'I' :
                case 'O' :
                case 'U' :i=1;
                System.out.println(ch);
            }
        }
        if(i==0)
            System.out.println("There are no vowels in a string");
    }
}
```

3140705 – Object Oriented Programming -I

Output:

The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac stringUowels.java

E:\darshan college\Java 2019-20\solution\javaprograms>java stringUowels
Enter a string : Computer Engineering
Uowels in a string are:
o
u
e
E
i
e
e
i

E:\darshan college\Java 2019-20\solution\javaprograms>
```

4. WAP that prompts the user to enter a decimal number and displays the number in fraction.

Hint: Read the decimal number as a string, extract the integer part and fractional part from the string.

```
import java.util.*;
import java.lang.Math;

//Class StringFraction
class StringFraction{
    //Declare Variables
    String strnum;
    int count;
    int number;
    int exponent;
    int numerator;
    int denominator;
    Scanner sc = new Scanner(System.in);

    void getStringNumber(){
        System.out.print("Enter number as string : ");
        strnum = sc.next();
    }
```

3140705 – Object Oriented Programming -I

```
void convert() {
    for(int i = 0; i < strnum.length() ;
        i++){if(strnum.charAt(i) == '.'){
            count = strnum.length() - (i+1);
            break;
        }
    }

    number      =      (int)(Double.parseDouble(strnum) * Math.pow(10,count));
    exponent = (int)Math.pow(10,count);
    System.out.println("Number : "+number);

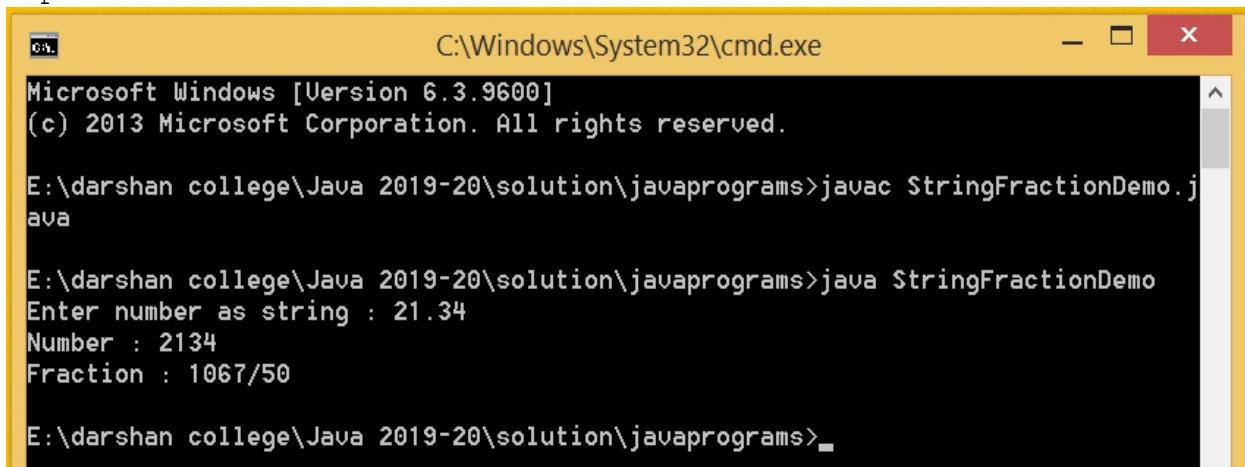
    //Getting GCD
    int gcd = 1, flag = 0;
    for(int i = 2; i<=number || i<=exponent;
        i++) {for(int j = 2; j < i; j++){
            if(i % j == 0)
                flag = 1;
        }
        if(flag == 0 && number % i == 0 && exponent % i ==
           0){gcd = gcd * i;
        }
        flag = 0;
    }

    //Actual Fraction
    numerator = number / gcd;
    denominator = exponent / gcd;
    System.out.println("Fraction : " + numerator + "/" +
denominator);
}

//Main Class
class StringFractionDemo{
    //Main Method
    public static void main(String[] args)
    { StringFraction sf = new
StringFraction();sf.getStringNumber();
    sf.convert();
}
}
```

3140705 – Object Oriented Programming -I

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac StringFractionDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java StringFractionDemo
Enter number as string : 21.34
Number : 2134
Fraction : 1067/50

E:\darshan college\Java 2019-20\solution\javaprograms>
```

3140705 – Object Oriented Programming -I

Practical-12

1. WAP that prompts the user to enter 5 numbers, stores them in an ArrayList, and displays them in increasing order.

```
import java.util.Scanner;
import java.util.ArrayList;

class Program1 {

    public static void main(String[] args) {
        // Create a Scanner
        Scanner sc = new Scanner(System.in);

        // Create an ArrayList of Integers
        ArrayList<Integer> list = new ArrayList<Integer>();

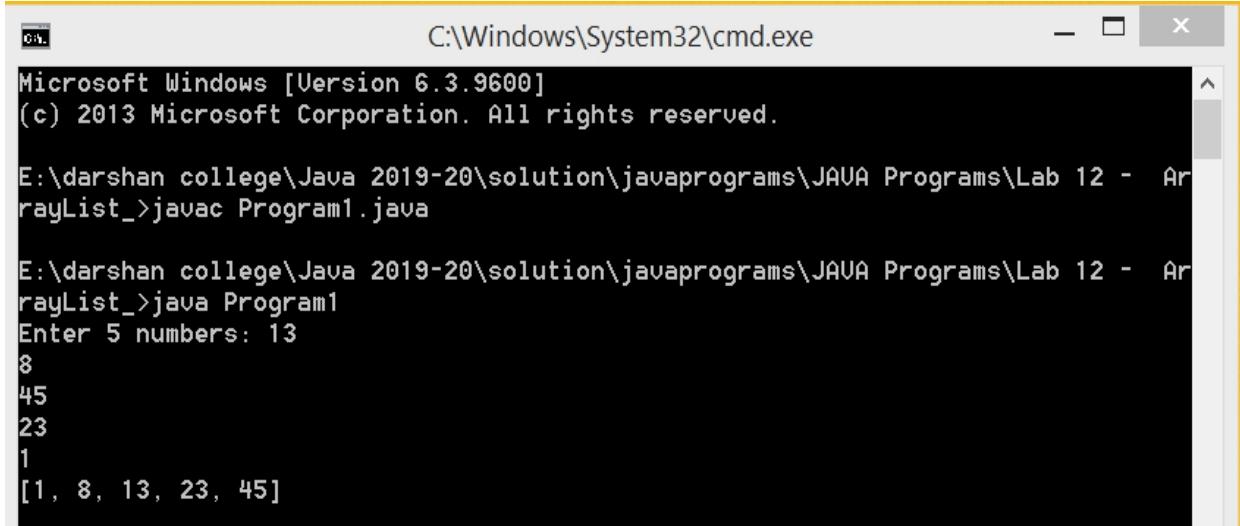
        // Prompt the user to enter 5 numbers
        System.out.print("Enter 5 numbers: ");
        for (int i = 0; i < 5; i++) {
            list.add(sc.nextInt());
        }
        // Sort numbers
        sort(list);

        // Display list in increasing order
        System.out.println(list.toString());
    }

    /**
     * Sorts an ArrayList of Integers */
    public static void sort(ArrayList<Integer> list)
        {java.util.Collections.sort(list);
    }
}
```

3140705 – Object Oriented Programming -I

Output:



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList>javac Program1.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList>java Program1
Enter 5 numbers: 13
8
45
23
1
[1, 8, 13, 23, 45]
```

2. WAP that creates ArrayList with following value:

“Delhi”, “Mumbai”, “Bangalore”, “Hyderabad” and “Ahmedabad”

Replace “Ahmedabad” with “Surat” in above ArrayList.

```
import java.util.ArrayList;
class Program2 {
    public static void main(String[] args) {

        // Create an ArrayList of City
        ArrayList<String> list = new ArrayList<>();

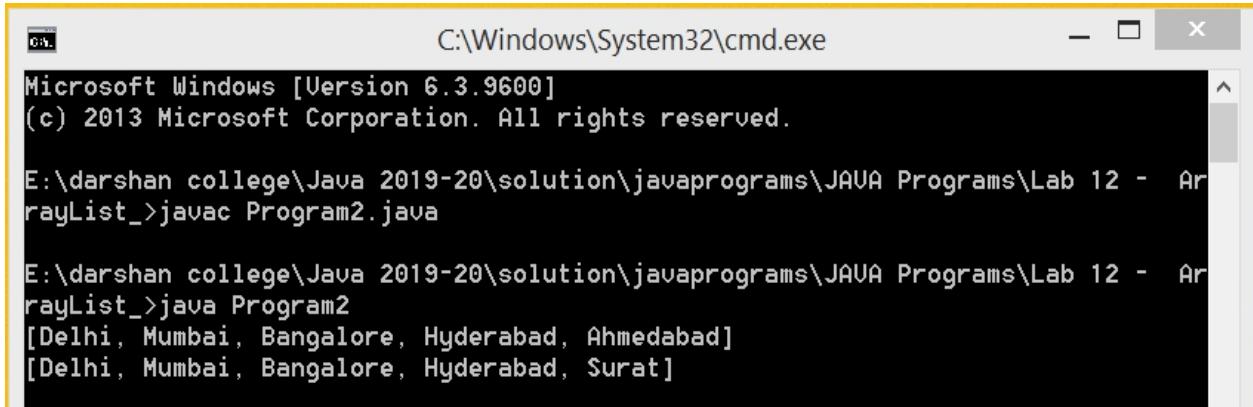
        list.add("Delhi");
        list.add("Mumbai");
        list.add("Bangalore");
        list.add("Hyderabad");
        list.add("Ahmedabad");
        System.out.println(list);

        // Replace "Ahmedabad" with "Surat" in above ArrayList.
        list.set(list.indexOf("Ahmedabad"), "Surat");
        System.out.println(list);

    }
}
```

3140705 – Object Oriented Programming -I

Output:



A screenshot of a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window shows the following text:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList_>javac Program2.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList_>java Program2
[Delhi, Mumbai, Bangalore, Hyderabad, Ahmedabad]
[Delhi, Mumbai, Bangalore, Hyderabad, Surat]

3. WAP that creates ArrayList with following value:

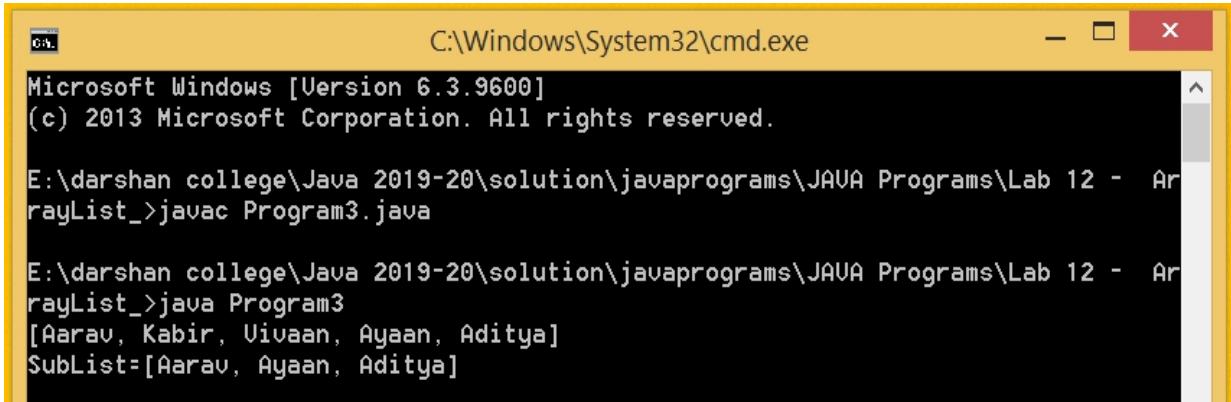
“Aarav”, “Kabir”, “Vivaan”, “Ayaan” and “Aditya”

Create a sublist with string from above ArrayList which start with ‘A’.

```
import java.util.ArrayList;  
  
class Program3 {  
    public static void main(String[] args) {  
  
        // Create an ArrayList of Name  
        ArrayList<String> list = new ArrayList<>();  
  
        list.add("Aarav");  
        list.add("Kabir");  
        list.add("Vivaan");  
        list.add("Ayaan");  
        list.add("Aditya");  
        System.out.println(list);  
  
        // Create a sublist with string from above ArrayList which strart with 'A'.  
        ArrayList<String> sublist = new ArrayList<>();  
        for (int i = 0; i < list.size(); i++) {  
            if (list.get(i).charAt(0) == 'A')  
                {sublist.add(list.get(i));}  
        }  
        System.out.println("SubList="+sublist);  
    }  
}
```

3140705 – Object Oriented Programming -I

Output:



A screenshot of a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window shows the following text output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList_>javac Program3.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 12 - ArrayList_>java Program3
[Aarav, Kabir, Vivaan, Ayaan, Aditya]
SubList=[Aarav, Ayaan, Aditya]
```

3140705 – Object Oriented Programming -I

Practical-13

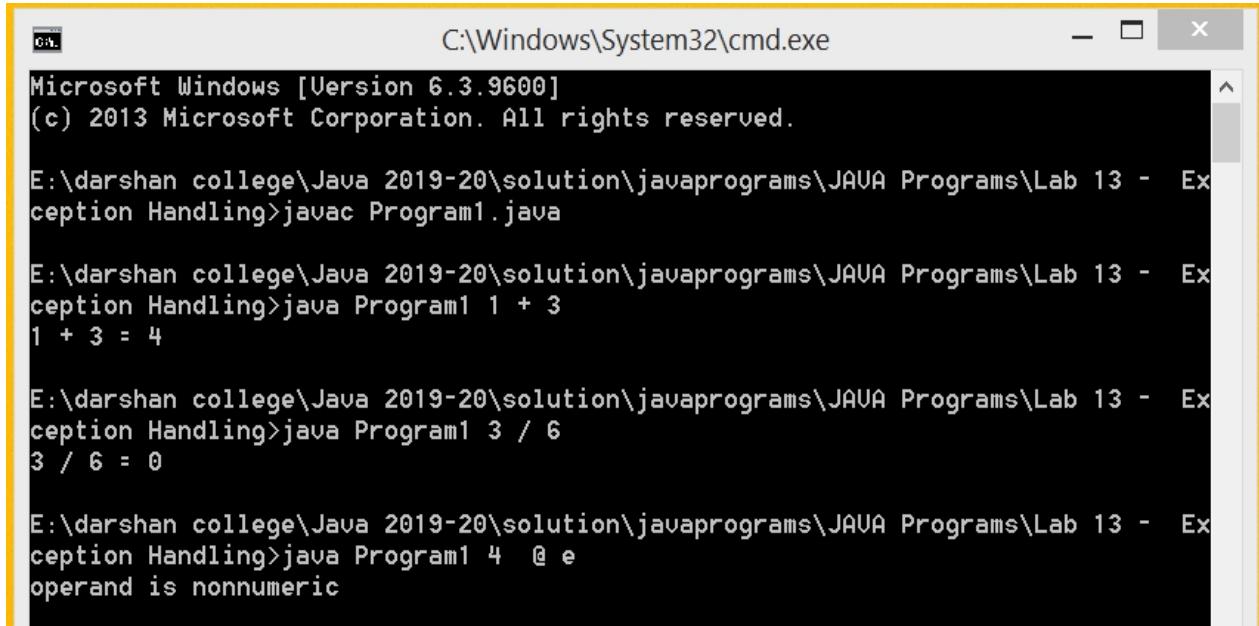
1. WAP to develop a simple command-line calculator which takes operand and operator as command-line argument, here program terminates if any operand is nonnumeric. exception handler to achieve the exception handling with nonnumeric operand and display message that informs the user of the wrong operand type before exiting.

```
class Program1 {  
    public static void main(String args[])  
    {int result = 0;  
    try {  
        //Scan operator and operand from command line argument  
        int firstNumber = Integer.parseInt(args[0]);  
        int secondNumber = Integer.parseInt(args[2]);  
        char operator = args[1].charAt(0);  
  
        // perform operation  
        switch (operator) {  
            case ('+'):  
                result = firstNumber + secondNumber;  
                System.out.println(args[0] + " " + args[1] + "  
" + args[2] + " = " + result);  
                break;  
  
            case ('-'):  
                result = firstNumber - secondNumber;  
                System.out.println(args[0] + " " + args[1] + "  
" + args[2] + " = " + result);  
                break;  
  
            case ('*'):  
                result = firstNumber * secondNumber;  
                System.out.println(args[0] + " " + args[1] + "  
" + args[2] + " = " + result);  
                break;  
  
            case ('/'):  
                result = firstNumber / secondNumber;  
                System.out.println(args[0] + " " + args[1] + "  
" + args[2] + " = " + result);  
                break;  
        }  
    } catch (Exception e) {  
        System.out.println("Error: Non-numeric operand");  
    }  
}
```

3140705 – Object Oriented Programming -I

```
        default:  
            System.out.print("Invalid Operator selected ");  
        }  
  
    } catch (ArithmException e)  
    { System.out.println(e.getMessage());  
    } catch (NumberFormatException e)  
    { System.out.println("operand is nonnumeric");  
    } catch (ArrayIndexOutOfBoundsException e)  
    { System.out.println("please enter operand");  
    }  
}  
}  
}
```

Output:



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>javac Program1.java  
  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>java Program1 1 + 3  
1 + 3 = 4  
  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>java Program1 3 / 6  
3 / 6 = 0  
  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>java Program1 4 @ e  
operand is nonnumeric
```

2. WAP to accept N integer numbers from the command line. Raise and handle exception following cases :

- when a number is -ve
- when a number is evenly divisible by 10
- when a number is greater than 1000 and less than 2000
- when a number is greater than 7000

Skip the number if an exception is raised for it, otherwise add it to find total sum.

```
class Program2 {  
    public static void main(String args[]) {
```

3140705 – Object Oriented Programming -I

```
int sum = 0;

for (int i = 0; i < args.length; i++) {

    // Scan number from Commandline
    int number = Integer.parseInt(args[i]);

    try {
        // a. - when a number is -ve
        if (number < 0) {
            throw new Exception("Number is Negative");
        }

        // b. - when a number is evenly divisible by 10
        if (number % 10 == 0) {
            throw new Exception("Number is evenly
                                divisible by 10");
        }

        // c. - when a number is greater than 1000 and
        // less than 2000
        if (number > 1000 && number < 2000) {
            throw new Exception("Number is greater than
                                1000 and less than 2000");
        }

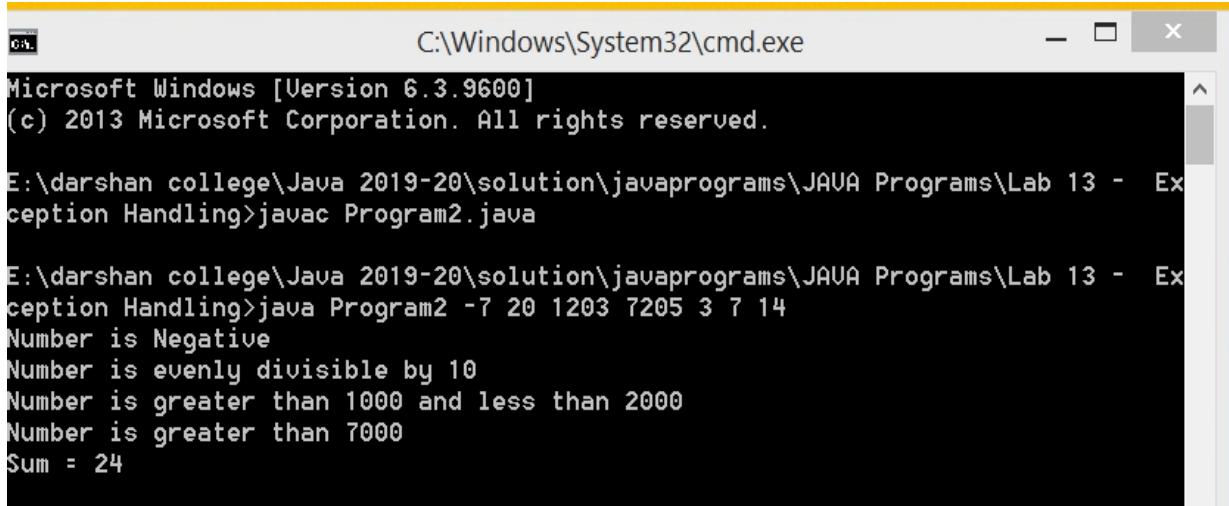
        // d. - when a number is greater than 7000
        if (number > 7000) {
            throw new Exception("Number is greater than
                                7000");
        }

        // add number
        sum = sum + number;
    } catch (Exception e)
        { System.out.println(e.getMessage()); }
    }

    System.out.println("Sum = " + sum);
}
```

3140705 – Object Oriented Programming -I

Output:



A screenshot of a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\System32\cmd.exe'. The window shows the following text:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>javac Program2.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>java Program2 -7 20 1203 7205 3 7 14
Number is Negative
Number is evenly divisible by 10
Number is greater than 1000 and less than 2000
Number is greater than 7000
Sum = 24

3. WAP to create Account class, which is representing a bank account where we can deposit and withdraw money. if we want to withdraw money which exceed our bank balance? will not be allowed, create InSufficientFundException to handle above situation and display proper error message.

```
// custom InSufficientFundException
class InSufficientFundException extends Exception
    {InSufficientFundException(String msg) {
        super(msg);
    }
}

class Account {
    double balance = 0;

    Account(int bal)
        { this.balance = bal;
    }

    void checkBalance() {
        System.out.println("Current balance" + balance);
    }

    void depositMoney(int bal)
        {if (bal > 0) {
            balance = balance + bal;
        }
    }
}
```

3140705 – Object Oriented Programming -I

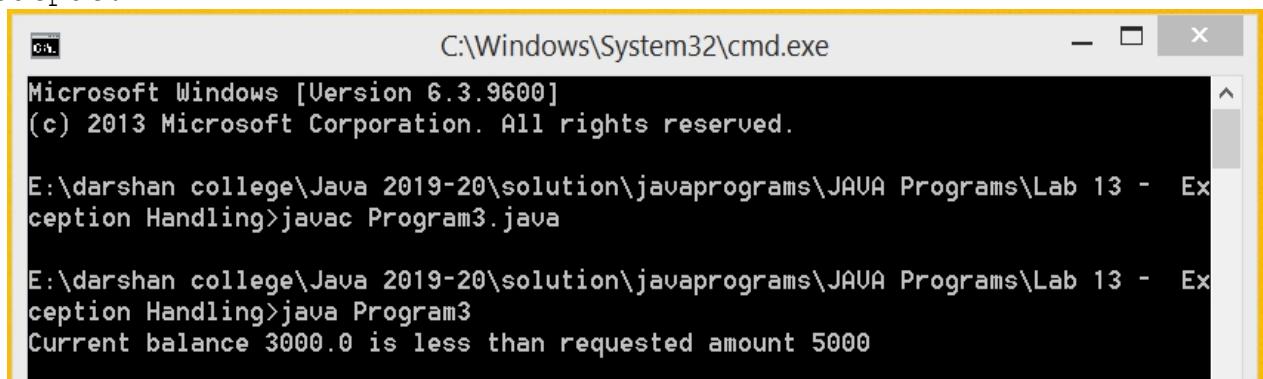
```
}

void withdrawMoney(int bal) throws InSufficientFundException
{
    if (bal > balance) {
        // throw custom InSufficientFundException exception
        throw new InSufficientFundException("Current balance "
            + balance + " is less than requested amount " + bal);
    }
    balance = balance - bal;
}

public class Program3 {
    public static void main(String args[])
    {Account a = new Account(3000);
     // handle InSufficientFundException
     try {
         a.withdrawMoney(5000);
     } catch (InSufficientFundException e) {

         System.out.println(e.getMessage());
     }
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>javac Program3.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 13 - Exception Handling>java Program3
Current balance 3000.0 is less than requested amount 5000
```

3140705 – Object Oriented Programming -I

Practical-14

1. The abstract Vegetable class has three subclasses named Potato, Brinjal and Tomato. Write a program that demonstrates how to establish this class hierarchy. Declare one instance variable of type String that indicates the color of a vegetable. Create and display instances of these objects. Override the `toString()` method of object to return a string with the name of vegetable and its color.

```
//Class Vegetable
abstract class Vegetable{
    String color;
    Vegetable(String color) {
        this.color = color;
    }
}

//Class Potato
class Potato extends Vegetable{
    Potato() {
        super("Cream");
    }
    public String toString() {
        return ("Potato : "+color);
    }
}

//Class Brinjal
class Brinjal extends Vegetable{
    Brinjal() {
        super("Purple");
    }
    public String toString() {
        return ("Brinjal : "+color);
    }
}

//Class Tomato
class Tomato extends Vegetable{
    Tomato() {
        super("Red");
    }
    public String toString() {
        return ("Tomato : "+color);
    }
}
```

3140705 – Object Oriented Programming -I

```

        }
    }

//Main Class
class VegetableDemo{
    //Main Method
    public static void main(String[] args)
    {Potato p = new Potato();
    Brinjal b = new Brinjal();
    Tomato t = new Tomato();
    //Display Vegetables with their color
    System.out.println(p.toString());
    System.out.println(b.toString());
    System.out.println(t.toString());
}
}

```

Output:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac VegetableDemo.java
E:\darshan college\Java 2019-20\solution\javaprograms>java VegetableDemo
Potato : Cream
Brinjal : Purple
Tomato : Red

```

2. Create interface EventListener with performEvent() method. Create MouseListener interface which inherits EventListener along with mouseClicked(), mousePressed(), mouseReleased(), mouseMoved(), mouseDragged() methods. Also create KeyListener interface which inherits EventListener along with keyPressed(), keyReleased() methods. WAP to create EventDe class which implements MouseListener and KeyListener and demonstrate all the methods of the interfaces.

```

interface EventListener{
    //performEvent Method
    void performEvent();
}

interface MouseListener extends EventListener{
    //mouseClicked Method
    void mouseClicked();
}

```

3140705 – Object Oriented Programming -I

```
//mousePressed Method
void mousePressed();
//mouseReleased Method
void mouseReleased();
//mouseMoved Method
void mouseMoved();
//mouseDragged Method
void mouseDragged();
}

interface KeyListener extends EventListener{
    //keyPressed Method
    void keyPressed();
    //keyReleased Method
    void keyReleased();
}

//Main Class
class EventDemo implements MouseListener, KeyListener{
    //performEvent Method
    public void performEvent(){
        System.out.println("Perform Event Method");
    }
    //mouseClicked Method
    public void mouseClicked(){
        System.out.println("Mouse Clicked");
    }
    //mousePressed Method
    public void mousePressed(){
        System.out.println("Mouse Pressed");
    }
    //mouseReleased Method
    public void mouseReleased(){
        System.out.println("Mouse Released");
    }
    //mouseMoved Method
    public void mouseMoved(){
        System.out.println("Mouse Moved");
    }
    //mouseDragged Method
    public void mouseDragged(){
        System.out.println("Mouse Dragged");
    }
    //keyPressed Method
    public void keyPressed(){

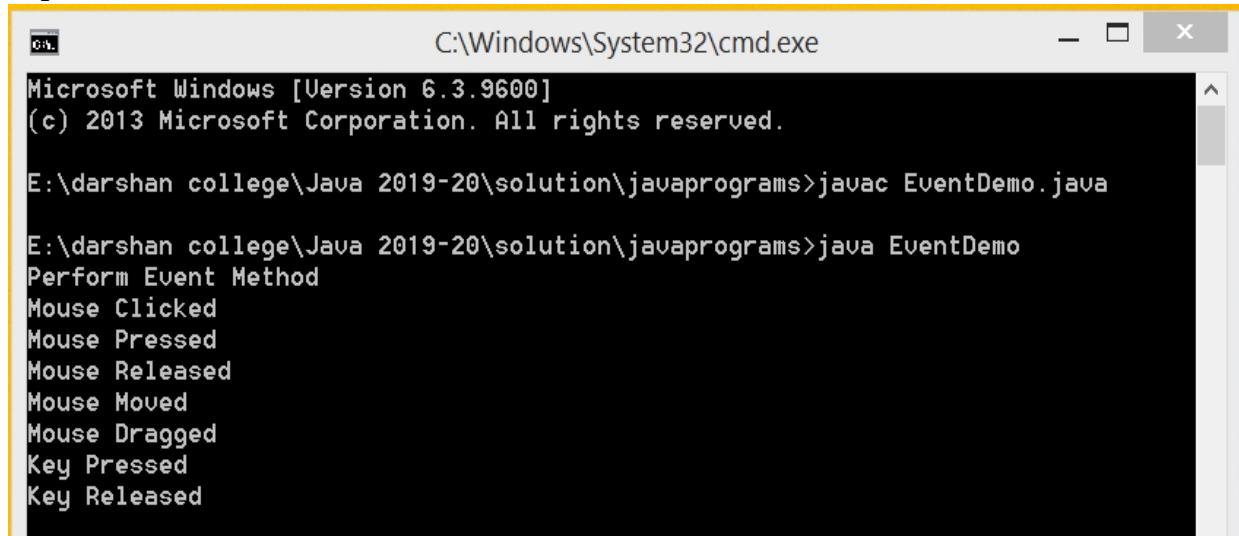
    }
}
```

3140705 – Object Oriented Programming -I

```
        System.out.println("Key Pressed");
    }
    //keyReleased Method
    public void keyReleased() {
        System.out.println("Key Released");
    }

    //Main Method
    public static void main(String[] args)
    {
        EventDemo e = new EventDemo();
        e.performEvent(); e.mouseClicked();
        e.mousePressed(); e.mouseReleased();
        e.mouseMoved();
        e.mouseDragged();
        e.keyPressed();
        e.keyReleased();
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac EventDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java EventDemo
Perform Event Method
Mouse Clicked
Mouse Pressed
Mouse Released
Mouse Moved
Mouse Dragged
Key Pressed
Key Released
```

3140705 – Object Oriented Programming -I

3. The Transport interface declares a deliver () method. The abstract class Animal is the super class of the Tiger, Camel, Deer and Donkey classes. The Transport interface is implemented by the Camel and Donkey classes. Write a ~~site~~ program that initialize an array of four Animal objects. If the object implements the Transport interface, the deliver () method is invoked.

```
interface
    Transport{void
        deliver();
    }
abstract class
    Animal{ abstract void
        display();
    }

//Tiger Class
class Tiger extends
    Animal{void display(){
        System.out.println("Tiger Class");
    }
}

//Camel Class
class Camel extends Animal implements
    Transport{void display(){
        System.out.println("Camel Class");
    }
    public void
        deliver(){ System.out.println("Camel
        deliver");
    }
}

//Deer Class
class Deer extends
    Animal{void display(){
        System.out.println("Deer Class");
    }
}

//Donkey Class
class Donkey extends Animal implements
    Transport{void display(){
        System.out.println("Donkey Class");
    }
}
```

Department of IT Engineering

Academic Year 2019 -20 | Semester-IV

Lab Solution

3140705 – Object Oriented Programming -I

```
    }
public void deliver() {
```

3140705 – Object Oriented Programming -I

```
        System.out.println("Donkey deliver");
    }
}

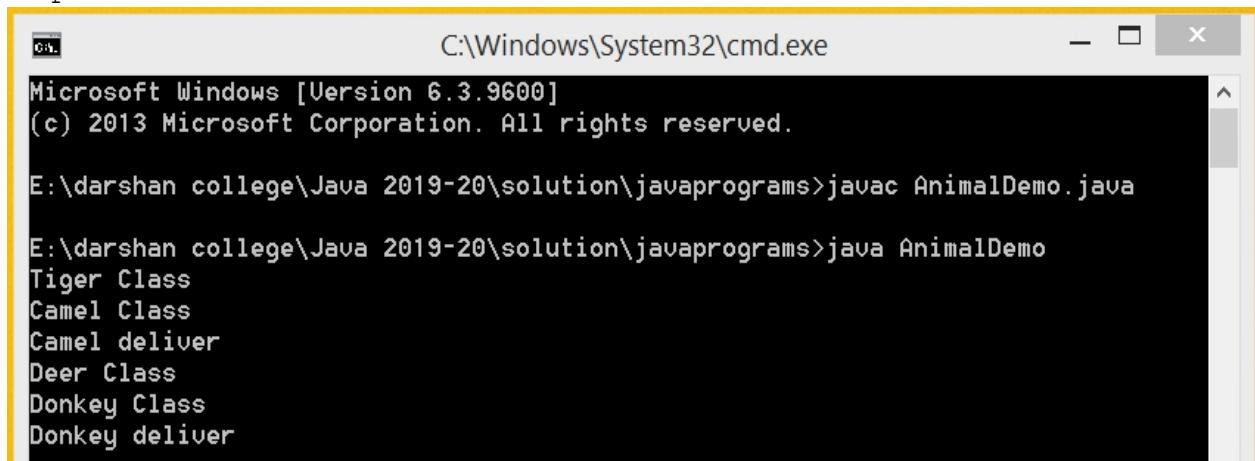
//Main Class
class AnimalDemo{
    public static void main(String[] args)
    {Tiger t = new Tiger();
     t.display();

     Camel c = new Camel();
     c.display();
     c.deliver();

     Deer d= new Deer();
     d.display();

     Donkey don= new Donkey();
     don.display();
     don.deliver();
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac AnimalDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java AnimalDemo
Tiger Class
Camel Class
Camel deliver
Deer Class
Donkey Class
Donkey deliver
```

3140705 – Object Oriented Programming -I

4. Declare a class called book having author_name as private data member. Extend book to have two sub classes called book_publication & paper_publication. Each of these classes have private member called title. Write a program to show usage of dynamic method dispatch (dynamic polymorphism) to display book or paper publications of given author. Use command line arguments for inputting data.

```
//class Book
class Book{
    private String authorName;

    Book(String a) {
        authorName = a;
    }
    void display() {
        System.out.println("Author : "+authorName);
    }
}

//class BookPublication
class BookPublication extends
Book{ private String title;
BookPublication(String a, String t) {
    super(a);
    title = t;
}
void display() {
    System.out.println("Book Title : "+title);
}
}

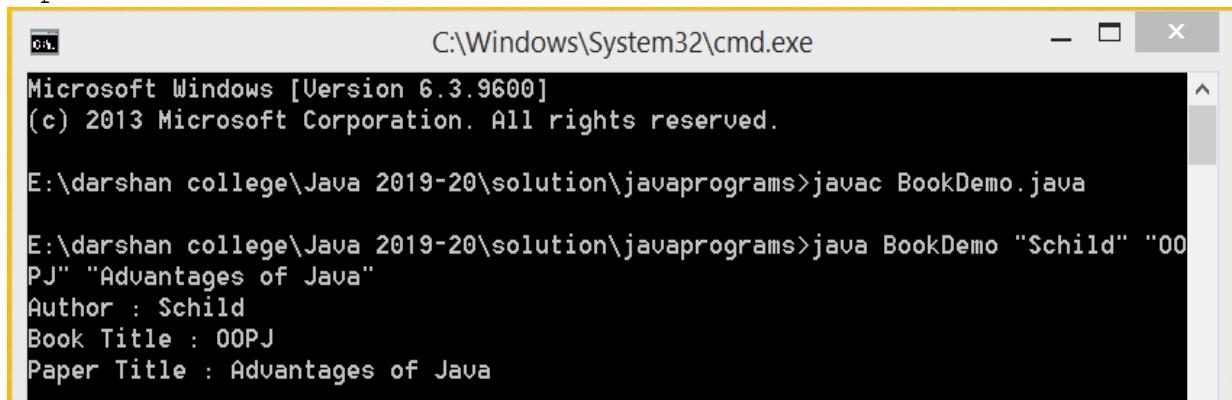
//class PaperPublication
class PaperPublication extends
Book{ private String title;
PaperPublication(String a, String t) {
    super(a);
    title = t;
}
void display() {
    System.out.println("Paper Title : "+title);
}
}
```

3140705 – Object Oriented Programming -I

```
//class Main class
class BookDemo{
    public static void main(String[] args)
        {Book b = new Book(args[0]);
        BookPublication bp = new BookPublication(args[0],args[1]);
        PaperPublication pp = new
PaperPublication(args[0],args[2]);

        Book bs = b;
        bs.display();
        bs = bp;
        bs.display();
        bs = pp;
        bs.display();
    }
}
```

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms>javac BookDemo.java

E:\darshan college\Java 2019-20\solution\javaprograms>java BookDemo "Schild" "OOPJ"
Author : Schild
Book Title : OOPJ
Paper Title : Advantages of Java
```

3140705 – Object Oriented Programming -I

Practical-15

1. Write a program to change the color of the circle from red to blue when mouse is click the circle.

```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.event.EventHandler;

import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;

import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class CircleDemo extends Application
{
    @Override
    public void start(Stage stage) {
        //Drawing a Circle
        Circle circle = new Circle();

        //Setting the position of the circle
        circle.setCenterX(300.0f);
        circle.setCenterY(135.0f);

        //Setting the radius of the circle
        circle.setRadius(25.0f);

        //Setting the color of the circle
        circle.setFill(Color.BROWN);

        //Setting the stroke width of the circle
        circle.setStrokeWidth(20);

        //Setting the text
        Text text = new Text("Click on the circle to change its
color");

        //Setting the font of the text
```

3140705 – Object Oriented Programming -I

```
text.setFont(Font.font(null, FontWeight.BOLD, 15));

//Setting the color of the text
text.setFill(Color.CRIMSON);

//setting the position of the text
text.setX(150);
text.setY(50);

//Creating the mouse event handler
EventHandler<MouseEvent> eventHandler = new EventHandler
<MouseEvent>()
{
    @Override
    public void handle(MouseEvent e)
    {
        System.out.println("Hello World");
        circle.setFill(Color.DARKSLATEBLUE);
    }
};
//Registering the event filter
circle.addEventFilter(MouseEvent.MOUSE_CLICKED,
eventHandler);

//Creating a Group object
Group root = new Group(circle, text);

//Creating a scene object
Scene scene = new Scene(root, 600, 300);

//Setting the fill color to the scene
scene.setFill(Color.LAVENDER);

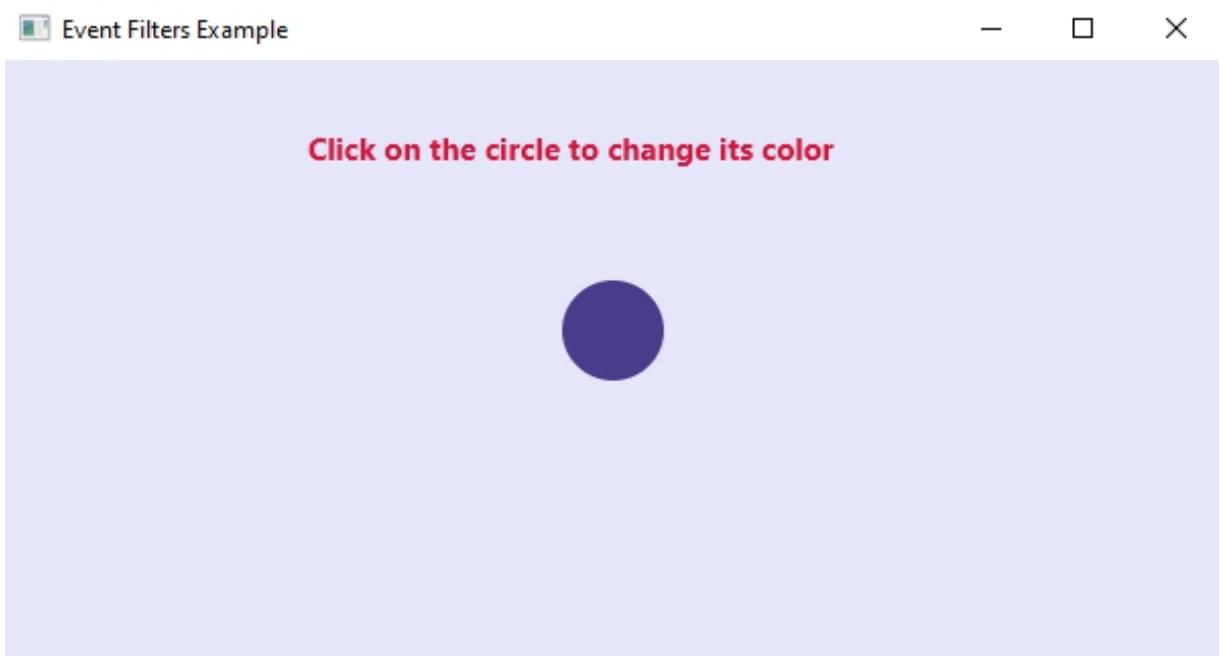
//Setting title to the Stage
stage.setTitle("Event Filters Example");

//Adding scene to the stage
stage.setScene(scene);

//Displaying the contents of the stage
stage.show();
}
public static void main(String
args[]) {launch(args);
}
```

3140705 – Object Oriented Programming -I

Output:



2. Write a program to detect and display the key pressed the keyboard.

```
import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.input.KeyEvent;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class KeyEvents extends Application
{
    public static void main(String[] args)
    {
        launch(args);
    }
    @Override
    public void start(Stage primaryStage)
    {
        Text text = new Text();
        text.setX(10.0);
        text.setY(100.0);
        text.setFont(Font.font("verdana",
            FontWeight.BOLD,
            FontPosture.REGULAR, 15));
    }
}
```

3140705 – Object Oriented Programming -I

```
Group root = new Group(text);
Scene scene = new Scene(root, 300, 250);
scene.setOnKeyPressed(new EventHandler<KeyEvent>() {
    public void handle(KeyEvent ke) {
        text.setText("Key      Pressed: " + ke.getCode().toString());
    }
});
primaryStage.setScene(scene);
primaryStage.setTitle("Press any key...");
primaryStage.show();
}
```

Output:



Key Pressed: CONTROL

3140705 – Object Oriented Programming -I

Practical-16

1. Write a program to design login screen.

```
import javafx.application.Application;
import javafx.event.*;
import static javafx.geometry.HPos.RIGHT;
import javafx.geometry.*;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.paint.Color;
import javafx.scene.text.*;
import javafx.stage.Stage;

public class Login extends Application

    {@Override
    public void start(Stage primaryStage)
        {primaryStage.setTitle("Login");
        GridPane grid = new GridPane();
        grid.setAlignment(Pos.CENTER);
        grid.setHgap(10); grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));

        Text scenetitle = new Text("Welcome");
        scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL,
        20));
        grid.add(scenetitle, 0, 0, 2, 1);

        Label userName = new Label("User Name:");
        grid.add(userName, 0, 1);

        TextField userTextField = new TextField();
        grid.add(userTextField, 1, 1);

        Label pw = new Label("Password:");
        grid.add(pw, 0, 2);

        PasswordField pwBox = new PasswordField();
        grid.add(pwBox, 1, 2);

        Button btn = new Button("Sign in");
```

3140705 – Object Oriented Programming -I

```
HBox hbBtn = new HBox(10);
hbBtn.setAlignment(Pos.BOTTOM_RIGHT);
hbBtn.getChildren().add(btn);
grid.add(hbBtn, 1, 4);

final Text actiontarget = new Text();
grid.add(actiontarget, 0, 6);
grid.setColumnSpan(actiontarget, 2);
grid.setAlignment(actiontarget, RIGHT);
actiontarget.setId("actiontarget");

btn.setOnAction(new EventHandler<ActionEvent>()

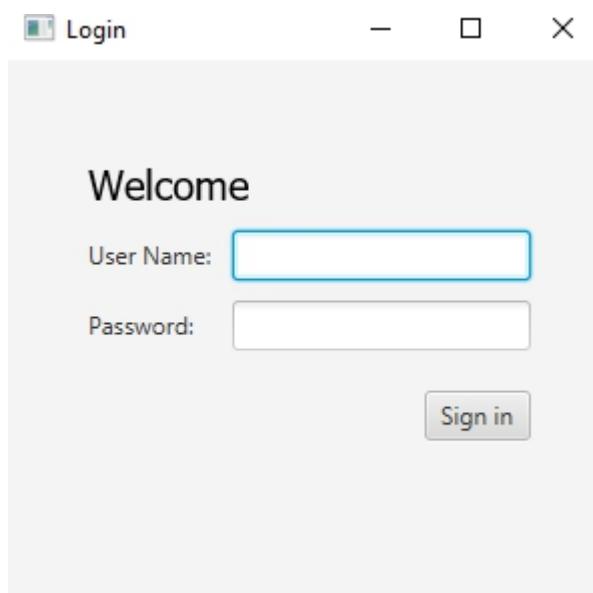
{
    @Override
    public void handle(ActionEvent e)
    {
        actiontarget.setFill(Color.FIREBRICK);
        actiontarget.setText("Sign in button pressed");
    }
});

Scene scene = new Scene(grid, 300, 275);
primaryStage.setScene(scene);
primaryStage.show();
}

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

3140705 – Object Oriented Programming -I

Output:



2. Write a program to design registration screen.

```
import javafx.application.Application;
import javafx.collections.*;
import javafx.geometry.*;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Registration extends Application
{
    @Override
    public void start(Stage stage) {
        //Label for name
        Text nameLabel = new Text("Name");

        //Text field for name
        TextField nameText = new TextField();

        //Label for date of birth
        Text dobLabel = new Text("Date of birth");

        //date picker to choose date
        DatePicker datePicker = new DatePicker();
```

3140705 – Object Oriented Programming -I

```
//Label for gender
Text genderLabel = new Text("Gender");

//Toggle group of radio buttons
ToggleGroup groupGender = new ToggleGroup();
RadioButton maleRadio = new RadioButton("Male");
maleRadio.setToggleGroup(groupGender);
RadioButton femaleRadio = new RadioButton("Female");
femaleRadio.setToggleGroup(groupGender);

//Label for reservation
Text reservationLabel = new Text("Reservation");

//Toggle button for reservation
ToggleButton Reservation = new ToggleButton();
ToggleButton yes = new ToggleButton("Yes");
ToggleButton no = new ToggleButton("No");
ToggleGroup groupReservation = new ToggleGroup();
yes.setToggleGroup(groupReservation);
no.setToggleGroup(groupReservation);

//Label for technologies known
Text technologiesLabel = new Text("Technologies Known");

//check box for education
CheckBox javaCheckBox = new CheckBox("Java");
javaCheckBox.setIndeterminate(false);

//check box for education
CheckBox dotnetCheckBox = new CheckBox(".Net");
dotnetCheckBox.setIndeterminate(false);

//Label for education
Text educationLabel = new Text("Educational qualification");

//list View for educational qualification
ObservableList<String> names =
FXCollections.observableArrayList(
"Engineering", "MCA", "MBA", "Graduation", "MTECH", "Mphil",
"Phd");

ListView<String> educationListView = new ListView<String>
(names);

//Label for location
```

3140705 – Object Oriented Programming -I

```
Text locationLabel = new Text("location");

//Choice box for location
ChoiceBox locationchoiceBox = new ChoiceBox();
locationchoiceBox.getItems().addAll("Ahmedabad", "Vadodara",
"Rajkot", "Mumbai", "Bangalore");

//Label for register
Button buttonRegister = new Button("Register");

//Creating a Grid Pane
GridPane gridPane = new GridPane();

//Setting size for the pane
gridPane.setMinSize(500, 500);

//Setting the padding
gridPane.setPadding(new Insets(10, 10, 10, 10));

//Setting the vertical and horizontal gaps between the columns
gridPane.setVgap(5);
gridPane.setHgap(5);

//Setting the Grid alignment
gridPane.setAlignment(Pos.CENTER);

//Arranging all the nodes in the grid
gridPane.add(nameLabel, 0, 0);
gridPane.add(nameText, 1, 0);

gridPane.add(dobLabel, 0, 1);
gridPane.add(datePicker, 1, 1);

gridPane.add(genderLabel, 0, 2);
gridPane.add(maleRadio, 1, 2);
gridPane.add(femaleRadio, 2, 2);
gridPane.add(reservationLabel, 0, 3);
gridPane.add(yes, 1, 3);
gridPane.add(no, 2, 3);

gridPane.add(technologiesLabel, 0, 4);
gridPane.add(javaCheckBox, 1, 4);
gridPane.add(dotnetCheckBox, 2, 4);
```

3140705 – Object Oriented Programming -I

```
gridPane.add(educationLabel, 0, 5);
gridPane.add(educationListView, 1, 5);

gridPane.add(locationLabel, 0, 6);
gridPane.add(locationchoiceBox, 1, 6);

gridPane.add(buttonRegister, 2, 8);

//Setting the back ground color
gridPane.setStyle("-fx-background-color: BEIGE;");

//Creating a scene object
Scene scene = new Scene(gridPane);

//Setting title to the Stage
stage.setTitle("Registration Form");

//Adding scene to the stage
stage.setScene(scene);

//Displaying the contents of the stage
stage.show();
}

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

}
```

3140705 – Object Oriented Programming -I

Output:

Registration Form

Name	<input type="text"/>
Date of birth	<input type="text"/> <input type="button" value="Calendar"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Reservation	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Technologies Known	<input checked="" type="checkbox"/> Java <input type="checkbox"/> .Net
Educational qualification	<ul style="list-style-type: none">EngineeringMCAMBAGraduationMTECHMphilPhd
location	<input type="button" value="▼"/>
<input type="button" value="Register"/>	

3140705 – Object Oriented Programming -I

Practical-17

1. WAP that counts number of characters, words, and lines in a file. Use exceptions to check whether the file that is read exists or not.

```
import java.io.*;
import java.util.*;

public class Program1 {
    public static void main(String args[]) throws Exception
    {int char_count = 0;
     int word_count = 0;
     int line_count = 0;
     String filename;
     String line;
     StringTokenizer st;

     Scanner sc = new Scanner(System.in);
     System.out.print("Enter filename : ");
     // Stored the entered file name in variable

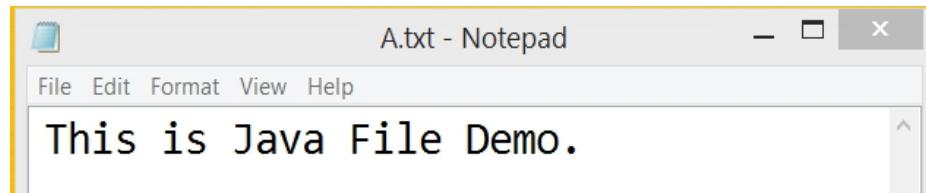
     filename = sc.nextLine();
     BufferedReader buf = new BufferedReader(new FileReader
     (filename));

     // read file line by line
     while ((line = buf.readLine()) != null)
         {line_count++; // for no. of lines
          st = new StringTokenizer(line);
          while (st.hasMoreTokens())
              {word_count++; // for no. of words
               String word = st.nextToken();
               char_count += word.length();
               // for no. of character

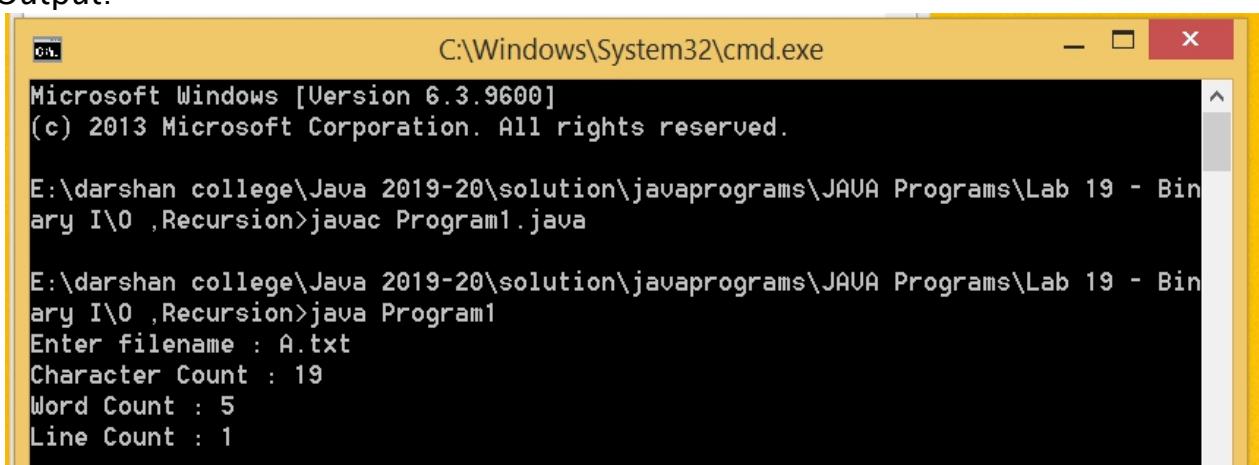
              }
         }
     System.out.println("Character Count : " + char_count);
     System.out.println("Word Count : " + word_count);
     System.out.println("Line Count : " + line_count);
     buf.close();
    }
}
```

3140705 – Object Oriented Programming -I

Input File “A.txt”:



Output:



2. WAP to replace all “word1” by “word2” from a file1, and output is written to file2 file and display the no. of replacement.

```

import java.io.*;
import java.util.*;

public class Program2 {
    public static void main(String args[]) throws Exception
    {int count = 0;
    File f = new File("A.txt");

    FileWriter writer = new FileWriter("B.txt");
    BufferedReader reader = new BufferedReader(new
    FileReader(f));
    String line = "", oldtxt = "";

    // read file line by line
    while ((line = reader.readLine()) != null)
        {System.out.println("line=" + line);
        // divide line into words
        StringTokenizer words = new StringTokenizer((line));
        oldtxt += line + "\n";
    }
}

```

3140705 – Object Oriented Programming -I

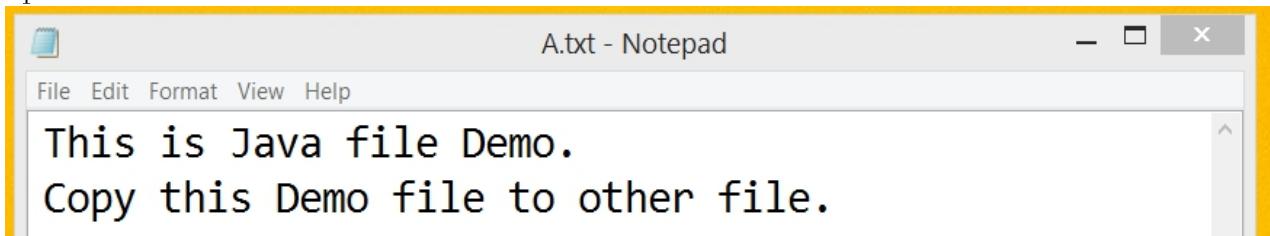
```
// Count number of replacement
while (words.hasMoreTokens()) {
    if (words.nextToken().equals("file"))
        count++;
}
}

// replace an old word with a new word
String newtxt = oldtxt.replace("file", "txtFile");
writer.write(newtxt);
System.out.println("\n\tReplacement:" + count);

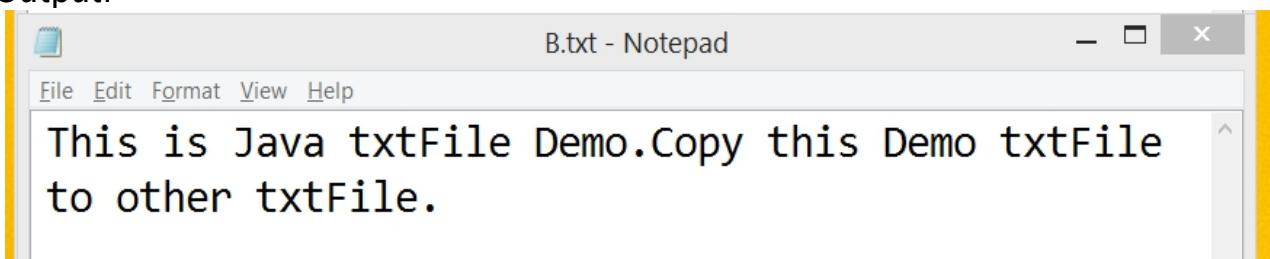
reader.close();
writer.flush();
writer.close();

}
}
```

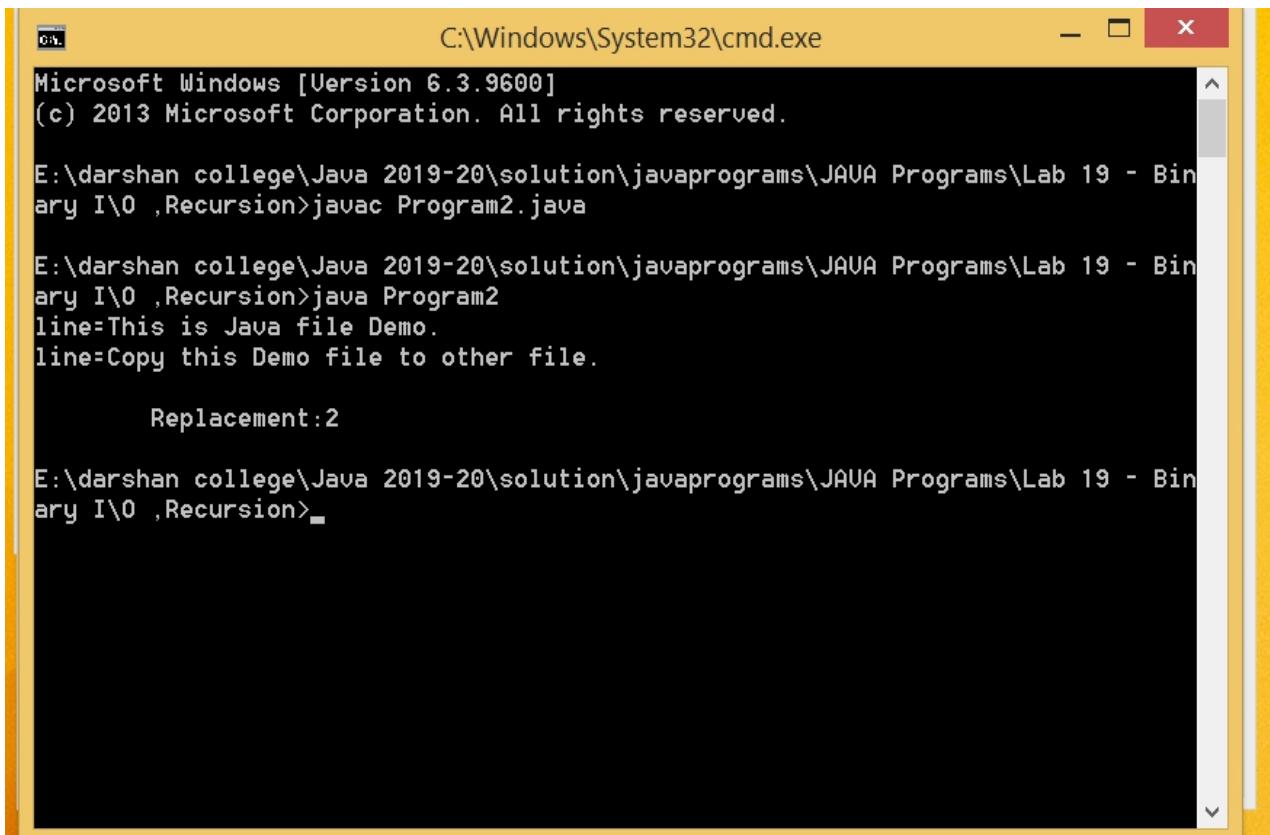
Input File “A.txt”:



Output:



3140705 – Object Oriented Programming -I



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>javac Program2.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>java Program2
line=This is Java file Demo.
line=Copy this Demo file to other file.

Replacement:2

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>
```

3. WAP to find factorial of given number using recursion.

```
import java.util.Scanner;

public class Program3 {
    public static void main(String args[]) throws Exception
        {Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number:");
        // Stored the entered value in variable
        int num = scanner.nextInt();

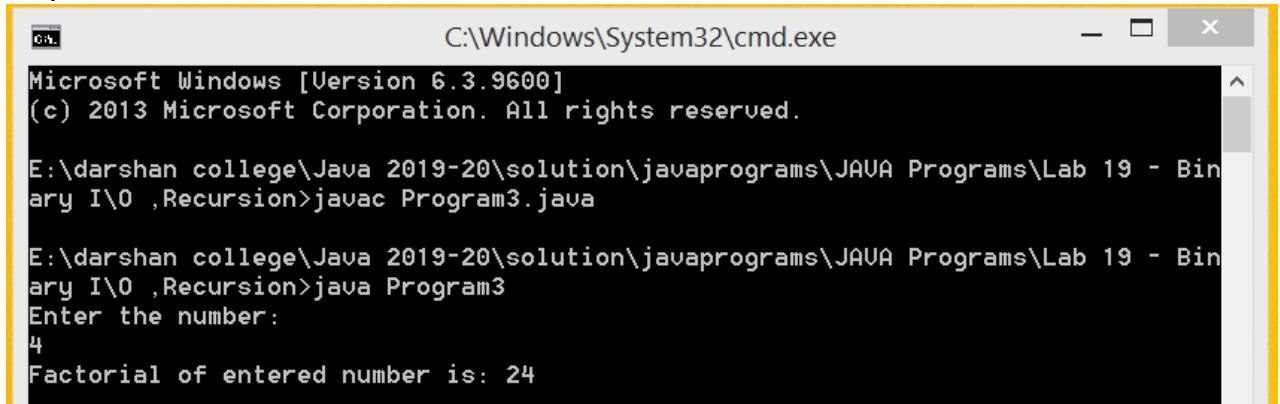
        // Called the user defined function fact
        int factorial = fact(num);
        System.out.println("Factorial of entered number is: " +
        factorial);
    }

    static int fact(int n)
    {int output;
    if (n == 1) {
```

3140705 – Object Oriented Programming -I

```
        return 1;
    }
    // Recursion: Function calling itself!!
    output = fact(n - 1) * n;
    return output;
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>javac Program3.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>java Program3
Enter the number:
4
Factorial of entered number is: 24
```

4. WAP to print Fibonacci for given number of terms using recursion.

```
import java.util.Scanner;

public class Program4 {
    static int n1 = 0, n2 = 1, n3 = 0;

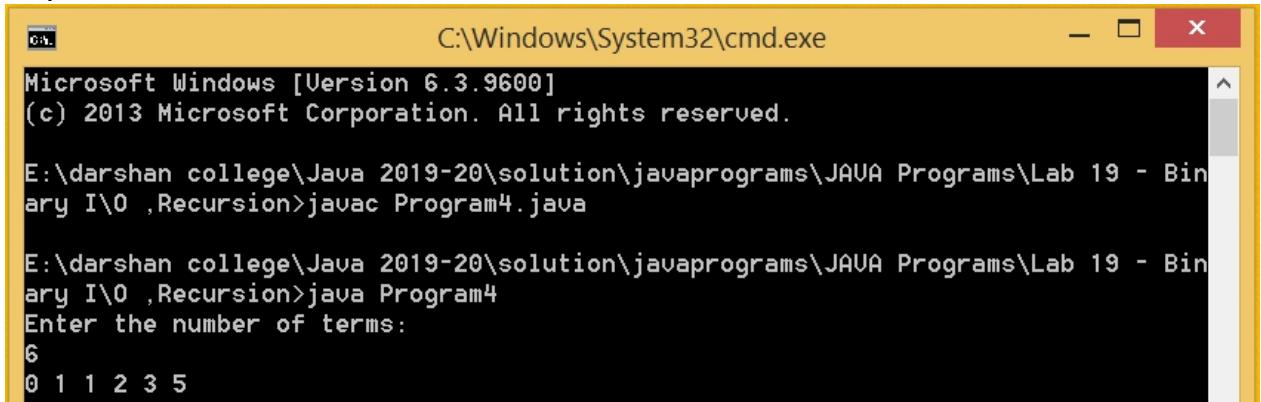
    public static void main(String args[]) throws Exception
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of terms:");
        // Stored the entered value in variable
        int count = scanner.nextInt();
        System.out.print(n1 + " " + n2);
        // Called the user defined function fibonacci
        fibonacci(count - 2);
    }

    static void fibonacci(int count)
    {
        if (count > 0) {
            n3 = n1 + n2;
            n1 = n2;
            n2 = n3;
            System.out.print(" " + n3);
        }
    }
}
```

3140705 – Object Oriented Programming -I

```
// Recursion: Function calling itself!!
fibonacci(count - 1);
}
}
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I\0 ,Recursion>javac Program4.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I\0 ,Recursion>java Program4
Enter the number of terms:
6
0 1 1 2 3 5
```

5. WAP that reads a file and counts the number of occurrences of digit enter by user. Supply the file name as a commandline argument.

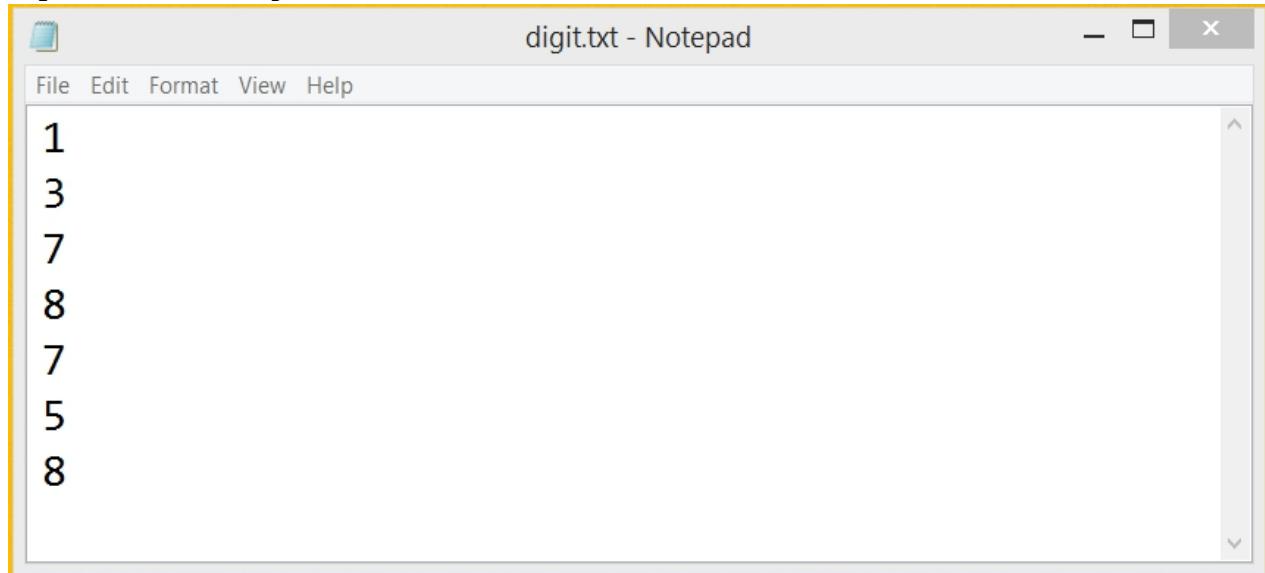
```
import java.io.*;
import java.util.Scanner;

public class Program5 {
    public static void main(String[] args) throws Exception
        {int count = 0, i;
        char ch;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the digit you want to search:");
        // Stored the entered value in variable
        ch = sc.next().charAt(0);
        // File Object Created
        File f = new File(args[0]);
        BufferedReader br = new BufferedReader(new FileReader(f));
        // read file character by character
        while ((i = br.read()) != -1) {
            if (i == ch)
                {count++;
            }
        }
        System.out.println("\n\tCount = " + count);
    }
}
```

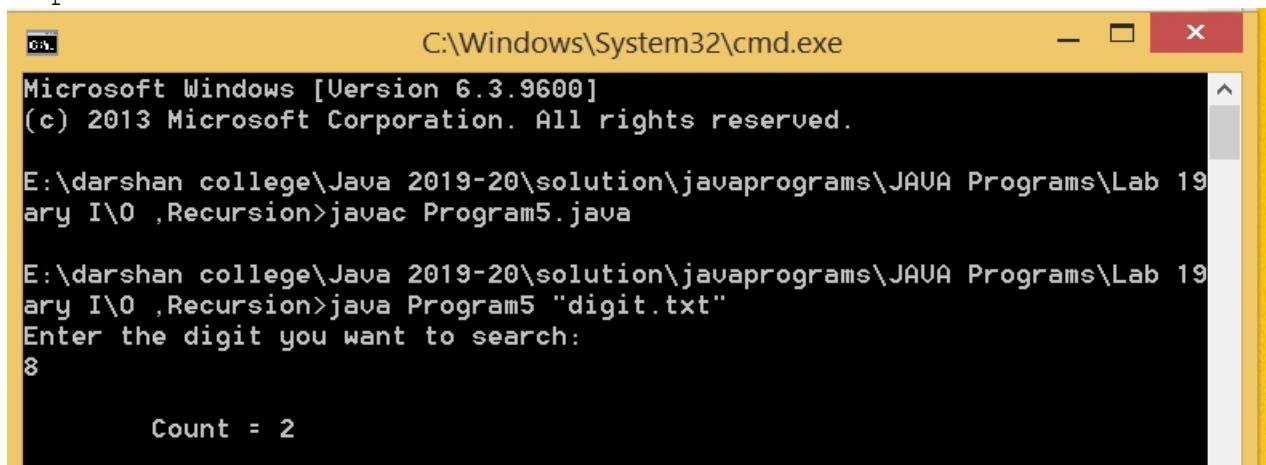
3140705 – Object Oriented Programming -I

```
}
```

Input File "digit.txt":



Output:



6. WAP to check that whether the name given from command line file or not? If it is a file then print the size of file and if it is directory then it should display the name of all files in it.

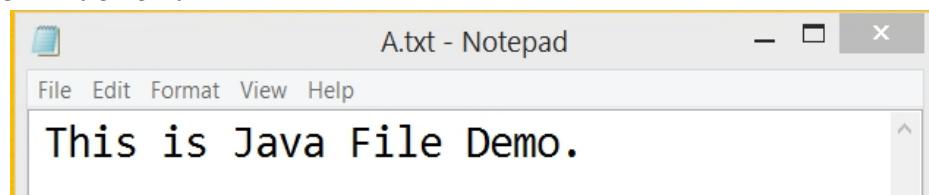
```
import java.io.*;

public class Program6 {
    public static void main(String[] args) throws Exception {
```

3140705 – Object Oriented Programming -I

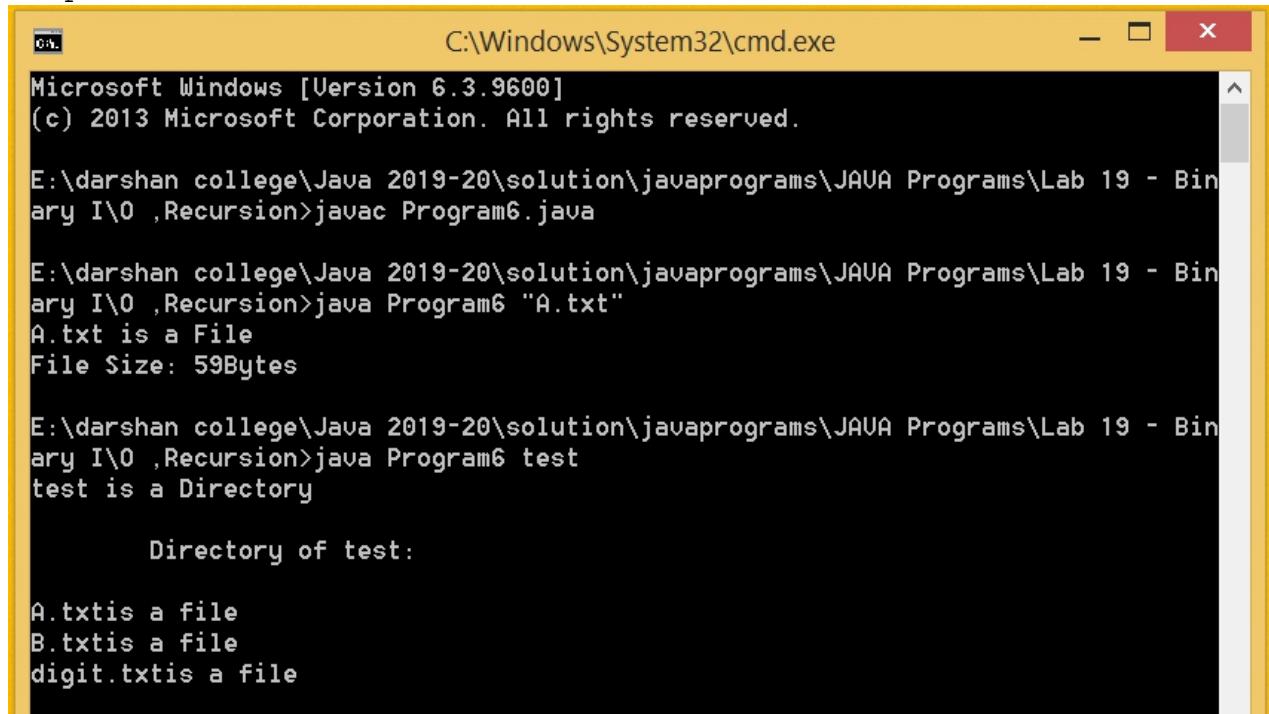
```
String dirname = args[0];
// File Object Created
File f1 = new File(dirname);
if (f1.isFile()) {
    System.out.println(dirname + " is a File");
    System.out.println("File Size: " + f1.length() +
    "Bytes");
} else if (f1.isDirectory())
{ System.out.println(dirname + " is a Directory");
System.out.println("\n\tDirectory of " + dirname +
":\n\t");
    // Stored the file and folder
    String s[] = f1.list();
    for (int i = 0; i < s.length; i++) {
        File f = new File(dirname + "/" + s[i]);
        if (f.isDirectory()) {
            System.out.println(s[i] + " is a directory");
        } else {
            System.out.println(s[i] + "is a file");
        }
    }
}
}
```

Input File “A.txt”:



3140705 – Object Oriented Programming -I

Output:



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I/O ,Recursion>javac Program6.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I\O ,Recursion>java Program6 "A.txt"
A.txt is a File
File Size: 59Bytes

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 19 - Binary I\O ,Recursion>java Program6 test
test is a Directory

        Directory of test:

A.txtis a file
B.txtis a file
digit.txtis a file
```

3140705 – Object Oriented Programming -I

Practical-18

1. Define generic class WildCard with method sum which add two generic values. Create class NumberDemo to demonstrate WildCard class.

```
import java.util.Arrays;
import java.util.List;

// generic class WildCard
class WildCard {
    public double sum(List<? extends Number> list)
        {double sum = 0.0;
        for (Number i : list) {
            sum += i.doubleValue();
        }

        return sum;
    }
}

class Program1 {
    public static void main(String args[])
    {WildCard w = new WildCard();
    // Upper Bounded Integer List
    List<Integer> list1 = Arrays.asList(4, 5, 6, 7);

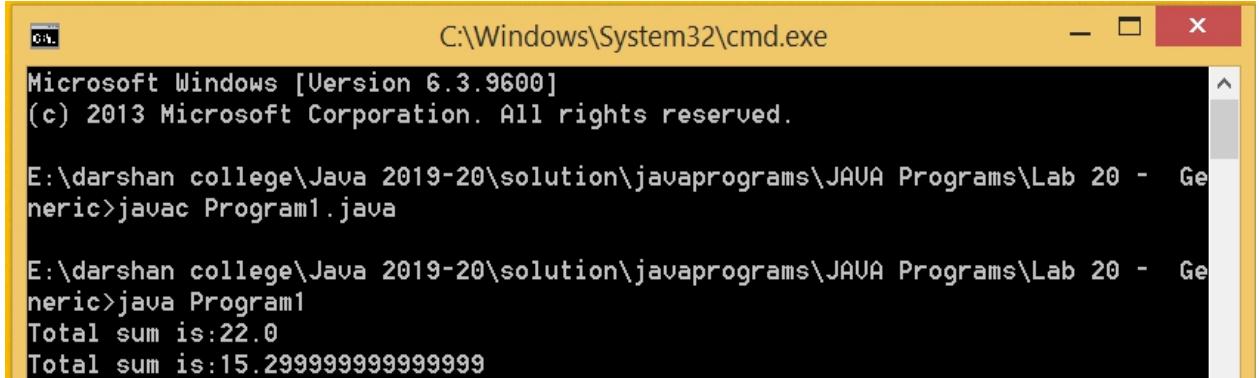
    // printing the sum of elements in list
    System.out.println("Total sum is:" + w.sum(list1));

    // Double list
    List<Double> list2 = Arrays.asList(4.1, 5.1, 6.1);

    // printing the sum of elements in list
    System.out.print("Total sum is:" + w.sum(list2));
}
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The output is as follows:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 20 - Generic>javac Program1.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 20 - Generic>java Program1
Total sum is:22.0
Total sum is:15.299999999999999
```

2. Implement the following method using linear search.

```
public static <E extends Comparable<E>>
int linearSearch(E[] list, E key)

import java.util.Scanner;

class Program2 {
    public static void main(String args[]) {
        Integer[] list = new Integer[10];
        Scanner sc = new Scanner(System.in);
        int search;
        for (int i = 0; i < list.length; i++)
            { System.out.print("list[" + i +
                "]=");list[i] = sc.nextInt();
        }
        System.out.println("Enter value to find");
        search = sc.nextInt();
        int foundAt = linearSearch(list, search);
        if (foundAt == -1) {
            System.out.println(search + " isn't present in
array.");
        } else {
            System.out.println(search + " is present at location "
+ foundAt + ".");
        }
    }

    // Public static <E extends Comparable<E>>Int linearsearch(E[]
list, E key)
    public static <E extends Comparable<E>> int linearSearch(E[]
list, E value) {
```

3140705 – Object Oriented Programming -I

```
for (int i = 0; i < list.length; i++)
    { if (list[i].compareTo(value) == 0)
        {
            return i;
        }
    }
return -1;
}
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 20 - Generic>javac Program2.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 20 - Generic>java Program2
list[0]=12
list[1]=34
list[2]=67
list[3]=89
list[4]=95
list[5]=100
list[6]=112
list[7]=134
list[8]=167
list[9]=223
Enter value to find
112
112 is present at location 6.
```

3140705 – Object Oriented Programming -I

Practical-19

1. Define **MyPriorityQueue** class that extends Priority Queue to implement the **Cloneable** interface and implement the **clone()** method to clone a priority queue.

```
import java.util.PriorityQueue;

//MyPriorityQueue class that extends Priority Queue to implement
//the Cloneable interface
class MyPriorityQueue<E> extends PriorityQueue<E> implements
Cloneable {

    // implement the clone() method
    public Object clone() throws CloneNotSupportedException {
        // clone a priority queue
        MyPriorityQueue<E> clone = new MyPriorityQueue<>();
        this.forEach(clone::offer);
        return clone;
    }
}

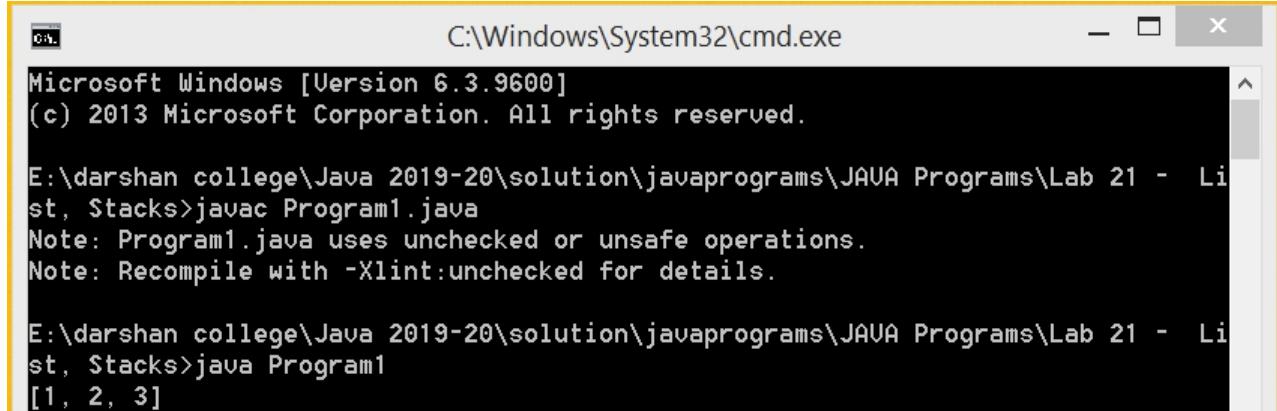
class Program1 {
    public static void main(String args[]) {
        // MyPriorityQueue object created.
        MyPriorityQueue<String> queue = new MyPriorityQueue<>();
        queue.offer("1");
        queue.offer("2");
        queue.offer("3");

        MyPriorityQueue<String> queue1 = null;
        try {
            queue1 = (MyPriorityQueue<String>) (queue.clone());
        } catch (CloneNotSupportedException e) {
            e.printStackTrace();
        }

        System.out.print(queue1);
    }
}
```

3140705 – Object Oriented Programming -I

Output:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The output is as follows:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 21 - List, Stacks>javac Program1.java
Note: Program1.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 21 - List, Stacks>java Program1
[1, 2, 3]
```

2. WAP to evaluate post-fix expression enter by command line arguments.

```
import java.util.Stack;

class Program2 {
    public static void main(String args[]) {
        if (args.length > 0)
            { String exp = args[0];
              System.out.println("postfix      evaluation:      " + 
evaluatePostfix(exp));
            } else {
              System.out.println("Enter expression");
            }
    }

    static int evaluatePostfix(String exp) {
        // create a stack
        Stack<Integer> stack = new Stack<>();

        // Scan all characters one by one
        for (int i = 0; i < exp.length(); i++)
            {char c = exp.charAt(i);

                // If the scanned character is an operand (number here),
                // push it to the stack.
                if (Character.isDigit(c))
                    stack.push(c - '0');

                // If the scanned character is an operator, pop two
                // elements from stack apply the operator
            }
    }
}
```

3140705 – Object Oriented Programming -I

```
else {
    int val1 = stack.pop();
    int val2 = stack.pop();

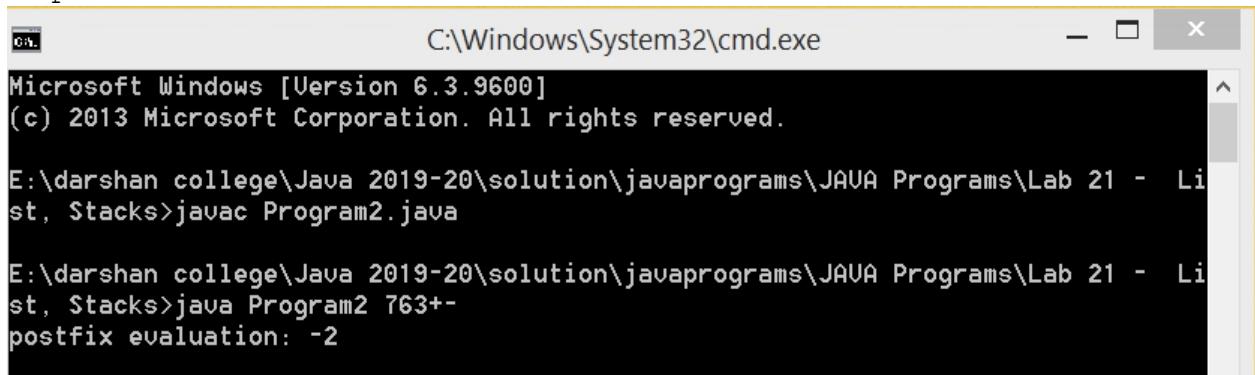
    switch (c) {
        case '+':
            stack.push(val2 + val1);
            break;

        case '-':
            stack.push(val2 - val1);
            break;

        case '/':
            stack.push(val2 / val1);
            break;

        case '*':
            stack.push(val2 * val1);
            break;
    }
}
return stack.pop();
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 21 - List, Stacks>javac Program2.java

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 21 - List, Stacks>java Program2 763+-
postfix evaluation: -2
```

3140705 – Object Oriented Programming -I

Practical-20

1. WAP to Create Set with following value:

```
{"London","New York","San Francisco","Beijing","New York"}
```

Perform following operation on above Set

- add new element
- get Set size
- remove element
- Contains element?
- addAll
- removeAll
- retainAll

```
import java.util.*;
class Program1 {
    public static void main(String args[]) {
        //set created
        Set hs = new HashSet();

        // - add new element
        hs.add("London");
        hs.add("New York");
        hs.add("San Francisco");
        hs.add("Beijing");
        hs.add("New York");

        // - get Set size
        System.out.println("Size= "+hs.size());

        // - remove element
        hs.remove("Beijing");
        System.out.println(hs);

        // - Contains element?
        System.out.println(hs.contains("Beijing"));

        // - addAll
        Set hs1 = new HashSet();
        hs1.add("Rajkot");
        hs1.add("Delhi");
        hs1.add("Goa");
        System.out.println("The items in set2 are "+hs1);
```

3140705 – Object Oriented Programming -I

```
        hs.addAll(hs1);
        System.out.println(hs);

        // - removeAll
        hs.removeAll(hs1);
        System.out.println(hs);

        // - retainAll
        hs1.add("New York");
        hs.retainAll(hs1);
        System.out.println(hs);
        System.out.println(hs1);
    }
}
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 22 - Sets and Maps>javac Program1.java
Note: Program1.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 22 - Sets and Maps>java Program1
Size= 4
[San Francisco, New York, London]
false
The items in set2 are [Delhi, Rajkot, Goa]
[San Francisco, New York, Delhi, Rajkot, London, Goa]
[San Francisco, New York, London]
[New York]
[Delhi, New York, Rajkot, Goa]
```

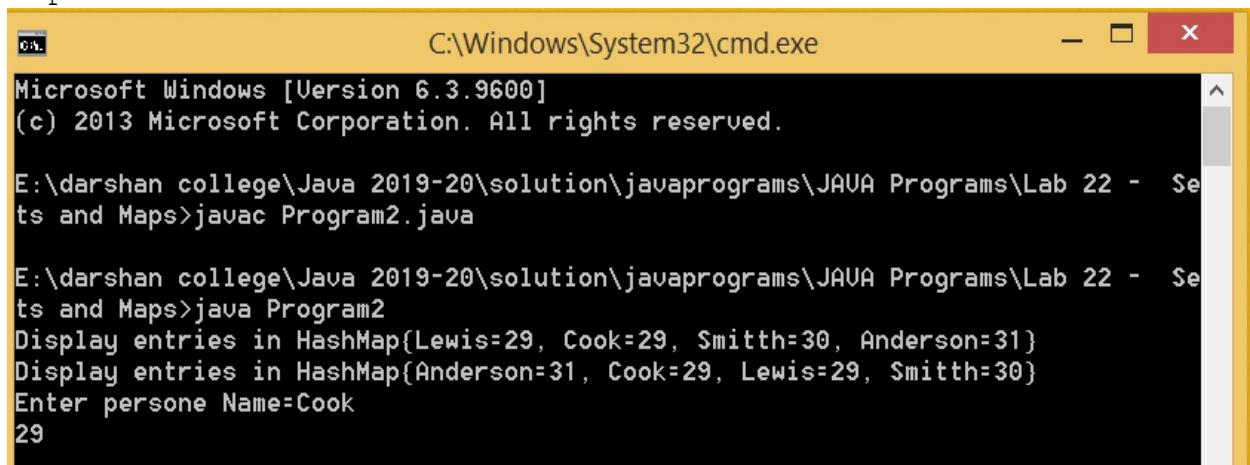
2. WAP to create Map<String, Integer> with following value
{"Smith",30}, {"Anderson",31}, {"Lewise",29}, {"Cook",29}
Perform following operation on above Map
 - Display entries in ascending order of key.
 - Display age of person entered by user.

```
import java.util.*;
class Program2 {
```

3140705 – Object Oriented Programming -I

```
public static void main(String args[]) {  
  
    Map<String, Integer>    hashMap    =    new    HashMap    <String,  
    Integer>();  
    hashMap.put("Smitth", 30);  
    hashMap.put("Anderson", 31);  
    hashMap.put("Lewis", 29);  
    hashMap.put("Cook", 29);  
    System.out.println("Display entries in HashMap"+hashMap);  
  
    // - Display entries in ascending order of key.  
    Map<String, Integer>    treeMap    =    new    TreeMap<String,  
    Integer>(hashMap);  
    System.out.println("Display entries in HashMap"+treeMap);  
  
    // - Display age of person entered by user.  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter persone Name=");  
    String name = sc.nextLine();  
    System.out.println(hashMap.get(name).intValue());  
  
}  
}
```

Output:



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 22 - Sets and Maps>javac Program2.java  
E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 22 - Sets and Maps>java Program2  
Display entries in HashMap{Lewis=29, Cook=29, Smitth=30, Anderson=31}  
Display entries in HashMap{Anderson=31, Cook=29, Lewis=29, Smitth=30}  
Enter persone Name=Cook  
29
```

3140705 – Object Oriented Programming -I

Practical-21

1. WAP to create two threads, one thread will print odd numbers and second thread will print even numbers between 1 to 1000 numbers.

```
class Thread1 extends Thread // implementing thread by class
{
    public Thread1()
    {
        start();
    }

    public void run() {
        for (int i = 1; i < 1000; i++)
            if (i % 2 != 0)
                System.out.println("odd Thread: " + i);
    }
}

class Thread2 implements Runnable // implementing thread by
interface
{
    Thread t;

    public Thread2() {
        t = new Thread(this);
        t.start();
    }

    public void run() {
        for (int i = 1; i < 1000; i++)
            if (i % 2 == 0)
                System.out.println("even Thread: " + i);
    }
}

public class Program1 {
    public static void main(String args[])
    {
        Thread1 t1=new Thread1();
        Thread2 t2=new Thread2();
    }
}
```

3140705 – Object Oriented Programming -I

Output:

```
odd Thread: 957
odd Thread: 959
odd Thread: 961
odd Thread: 963
odd Thread: 965
odd Thread: 967
odd Thread: 969
odd Thread: 971
odd Thread: 973
odd Thread: 975
odd Thread: 977
odd Thread: 979
odd Thread: 981
even Thread: 998
odd Thread: 983
odd Thread: 985
odd Thread: 987
odd Thread: 989
odd Thread: 991
odd Thread: 993
odd Thread: 995
odd Thread: 997
odd Thread: 999

E:\darshan college\Java 2019-20\solution\javaprograms\JAVA Programs\Lab 23 - Con
```

2. WAP to implement producer consumer problem.

```
class CI {
    int n;
    boolean valueSet = false;

    synchronized int receive()
    {while (!valueSet)
        try
            { wait()
            ;
        } catch (InterruptedException e)
            { System.out.println("InterruptedException caught");
        }
    System.out.println("Receive : " + n);
    valueSet = false;
    notify();
    return n;
}
```

3140705 – Object Oriented Programming -I

```
synchronized void send(int n)
    {while (valueSet)
        try
            { wait()
            ;
            } catch (InterruptedException e)
                { System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Send : " + n);
        notify();
    }
}

class Producer implements Runnable
{CI ci;

Producer(CI ci)
    {this.ci = ci;
    new Thread(this, "Producer").start();
}

public void run()
    {int i = 0;
    while (true) {
        ci.send(i++);
    }
}

class Consumer implements Runnable
{CI ci;

Consumer(CI ci)
    {this.ci = ci;
    new Thread(this, "Consumer").start();
}

public void run()
    {while (true) {
        ci.receive();
    }
}
```

Department of IT Engineering

Academic Year 2019 -20 | Semester-IV

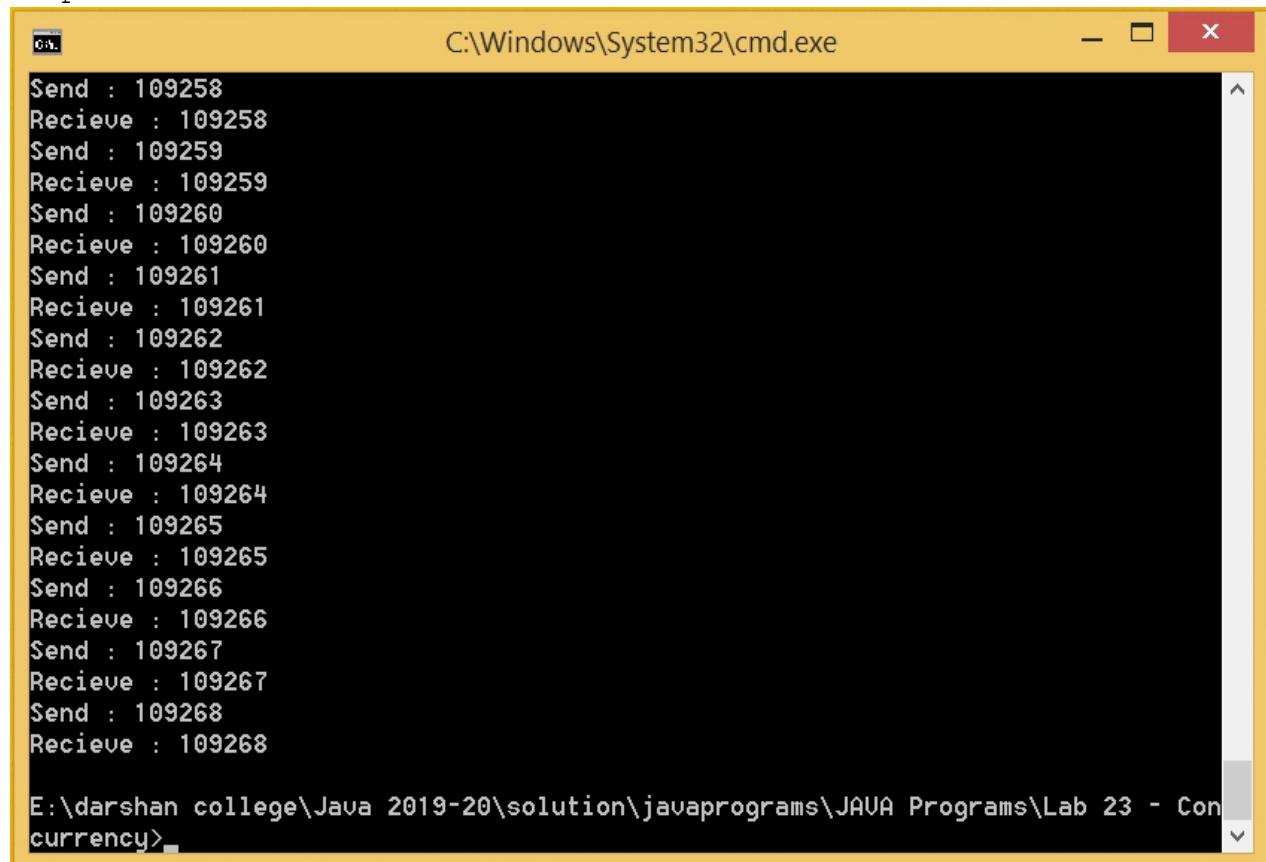
Lab Solution

3140705 – Object Oriented Programming -I
}

3140705 – Object Oriented Programming -I

```
public class Program2 {  
    public static void main(String args[])  
    {CI ci = new CI();  
     new Producer(ci);  
     new Consumer(ci);  
     System.out.println("Press Control-C to Stop.");  
  
    }  
}
```

Output:



```
Send : 109258  
Recieve : 109258  
Send : 109259  
Recieve : 109259  
Send : 109260  
Recieve : 109260  
Send : 109261  
Recieve : 109261  
Send : 109262  
Recieve : 109262  
Send : 109263  
Recieve : 109263  
Send : 109264  
Recieve : 109264  
Send : 109265  
Recieve : 109265  
Send : 109266  
Recieve : 109266  
Send : 109267  
Recieve : 109267  
Send : 109268  
Recieve : 109268  
  
E:\darshan college\Java 2019-20\solution\javaprograms\JAU A Programs\Lab 23 - Con  
currency>
```