Programming Lab Experiments

Junaid Girkar | 60004190057 | A3
Academic Year – 2020-21
S.E Computer Engineering

EXPERIMENT 1

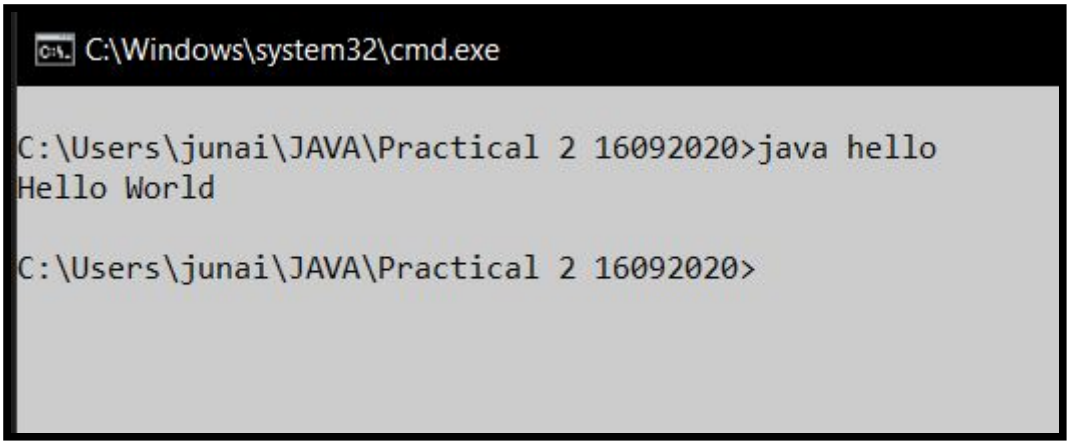
AIM: To implement Java Program Structures & Simple Programs

Q1. WAP to display hello Message on screen.

CODE:

```
import java.util.*;

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

OUTPUT:



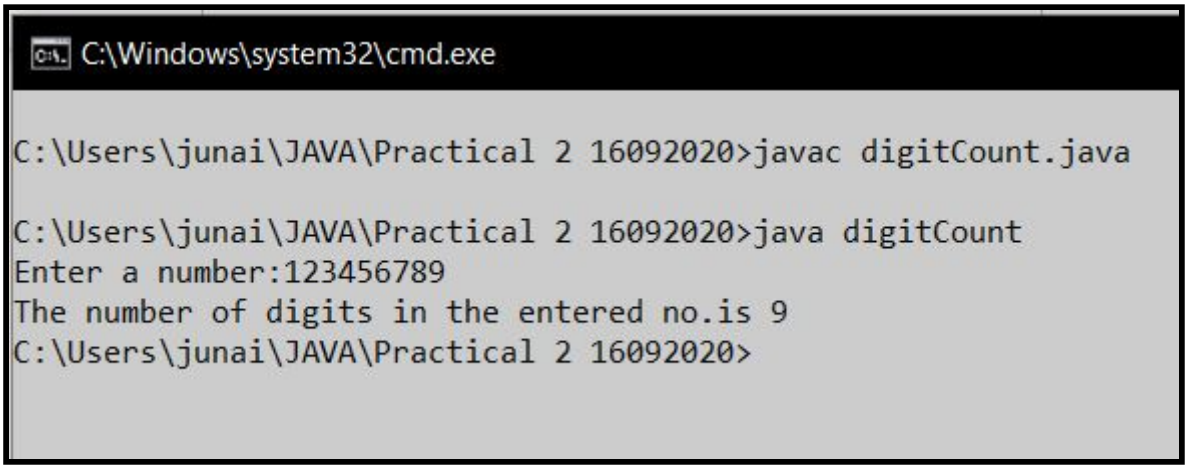
Q2. Write a Java program that reads a positive integer from the command line and count the number of digits the number (less than ten billion) has.

CODE:

```
import java.util.Scanner;
class digitCount
{
public static void main(String args[])
{
int n,count=0;
Scanner scan=new Scanner(System.in);
System.out.print("Enter a number:");
n=scan.nextInt();
while(n!=0)
{
n=n/10;
count++;
}
System.out.print("The number of digits in the entered no.is "+count);
}
```

```
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 2 16092020>javac digitCount.java
C:\Users\junai\JAVA\Practical 2 16092020>java digitCount
Enter a number:123456789
The number of digits in the entered no.is 9
C:\Users\junai\JAVA\Practical 2 16092020>

EXPERIMENT 2

AIM: To implement Java control statements and loops

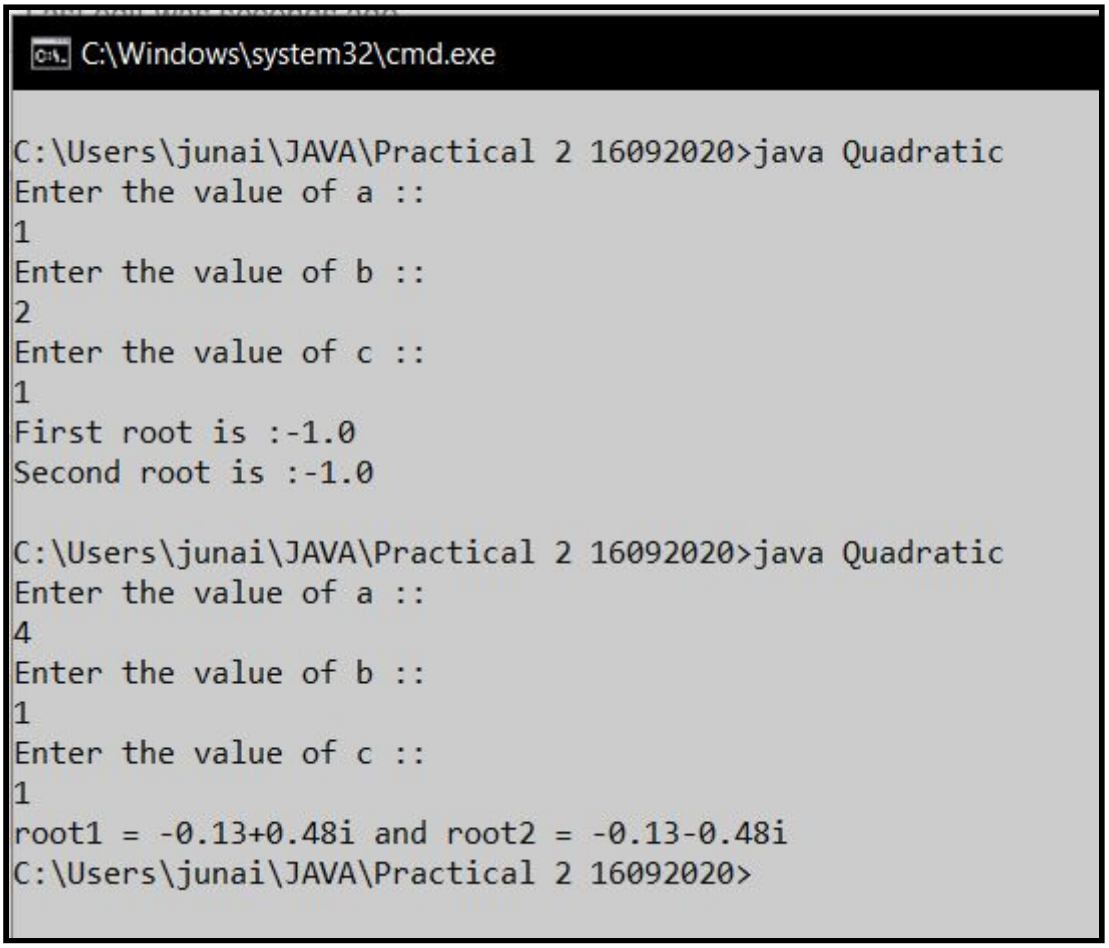
Q1. WAP to find roots of a Quadratic equation. Take care of imaginary values.

CODE:

```
import java.util.Scanner;
class Quadratic {
public static void main(String args[]){
double root1 = 0, root2 = 0;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the value of a ::");
double a = sc.nextDouble();
System.out.println("Enter the value of b ::");
double b = sc.nextDouble();
System.out.println("Enter the value of c ::");
double c = sc.nextDouble();
double determinant = (b*b)-(4*a*c);
double sqrt = Math.sqrt(determinant);
String sq=sqrt+"i";
if(determinant > 0) {
root1 = (-b + sqrt) / (2 * a);
root2 = (-b - sqrt) / (2 * a);
System.out.println("First root is :"+root1);
System.out.println("Second root is :"+root2);
}
else if (determinant == 0) {
root1 = -b / (2 * a);
root2=root1;
System.out.println("First root is :"+root1);
System.out.println("Second root is :"+root2);
}
else {
double realPart = -b / (2 *a);
double imaginaryPart = Math.sqrt(-determinant) / (2 * a);
System.out.format("root1 = %.2f+%.2fi and root2 = %.2f-%.2fi",
```

```
realPart, imaginaryPart, realPart, imaginaryPart);
}
}
}
```

OUTPUT:



Q2. Write a menu driven program using switch case to perform mathematical operations.

CODE:

```
import java.util.*;

class switchCase {
     public static void main(String args[]) {
     Scanner sc = new Scanner(System.in);
     System.out.println(
               "Select your operation:\n 1.Addition\n
2.Subtraction\n 3.Multiplication\n 4.Division\n ");
     int choice = sc.nextInt();
     System.out.println("Enter the first Number:");
     int num1 = sc.nextInt();
     System.out.println("Enter the second number:");
     int num2 = sc.nextInt();
     switch (choice) {
          case (1):
               int sum = num1 + num2;
               System.out.print("Sum : " + sum);
               break;
          case (2):
               int difference = num1 - num2;
                 System.out.println("Difference : " +
difference);
               break;

          case (3):
               int product = num1 * num2;
                 System.out.println("product : " +
product);
               break;

          case (4):
               int division = num1 / num2;
               System.out.println("Division : " +
division);
               break;
```

```
            default:
                System.out.println("Invalid Choice");
        }
    }
}
```

OUTPUT:

C:\Windows\system32\cmd.exe

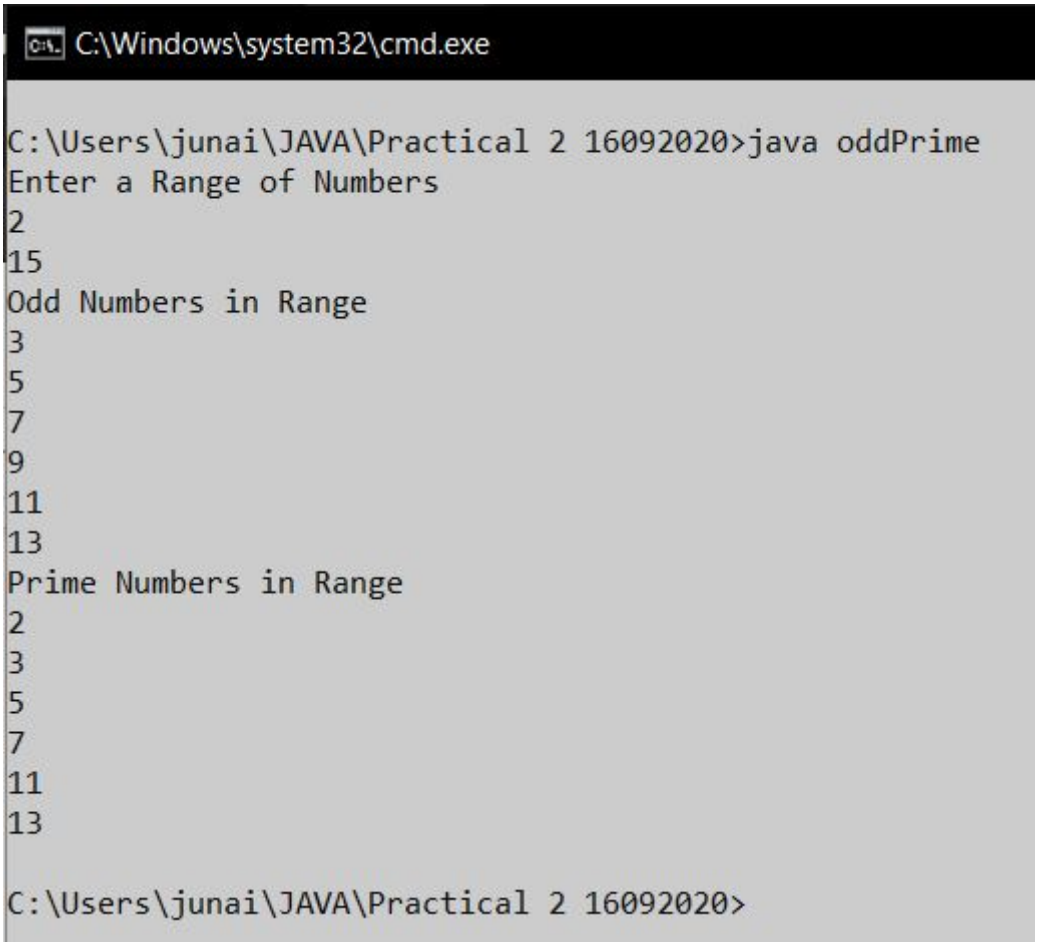
```
C:\Users\junai\JAVA\Practical 2 16092020>java switchCase
Select your operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division

1
Enter the first Number:
2
Enter the second number:
3
Sum : 5
C:\Users\junai\JAVA\Practical 2 16092020>java switchCase
Select your operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division

2
Enter the first Number:
2
Enter the second number:
3
Difference : -1

C:\Users\junai\JAVA\Practical 2 16092020>java switchCase
Select your operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division

3
Enter the first Number:
2
Enter the second number:
3
product : 6

C:\Users\junai\JAVA\Practical 2 16092020>java switchCase
Select your operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division
```

Q3. WAP to display odd numbers from given range/ prime numbers from given range.

CODE:

```
import java.util.Scanner;
class oddprime {
public static void main(String[] args) {
int count = 0;
Scanner ob = new Scanner(System.in);
System.out.println("Enter a Range of Numbers");
int beg = ob.nextInt();
int end = ob.nextInt();
System.out.println("Odd Numbers in Range");
for (int i = beg; i < end; i++) {
if (i % 2 == 1)
System.out.println(i);
}
System.out.println("Prime Numbers in Range");
for (int j = beg; j < end; j++) {
for (int k = 2; k < j; k++) {
if (j % k == 0) {
count++;
}
}
if (count == 0)
System.out.println(j);
count = 0;
}
ob.close();
}
}
```

OUTPUT:



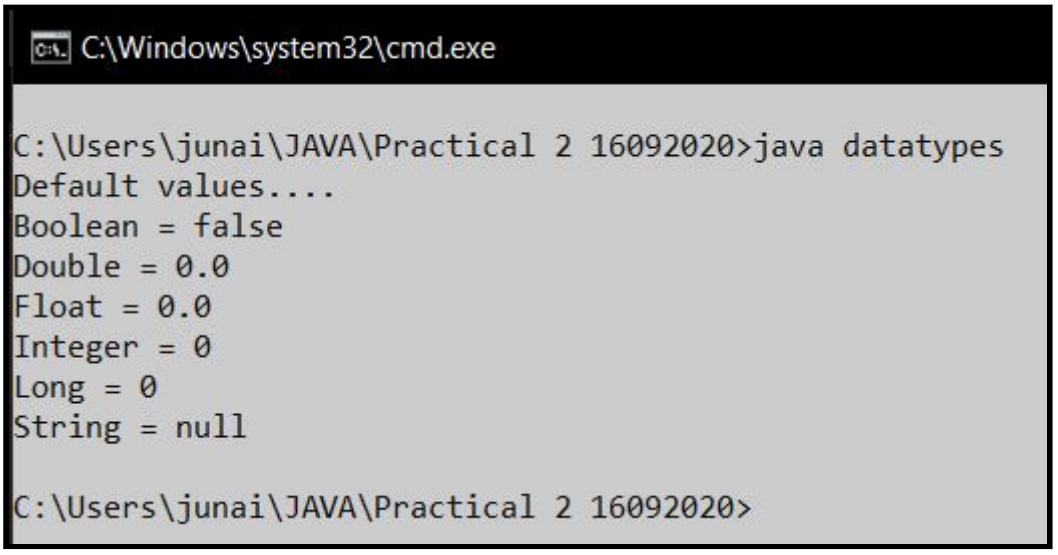
Q4. WAP to display default value of primitive data types.

CODE:

```
public class datatypes {
static boolean val1;
static double val2;
static float val3;
static int val4;
static long val5;
static String val6;
public static void main (String args[])
```

```
{
System.out.println("Default values....");
System.out.println("Val 1="+val1);
System.out.println("Val 2="+val2);
System.out.println("Val 3="+val3);
System.out.println("Val 4="+val4);
System.out.println("Val 5="+val5);
System.out.println("Val 6="+val6);
}
}
```

OUTPUT:



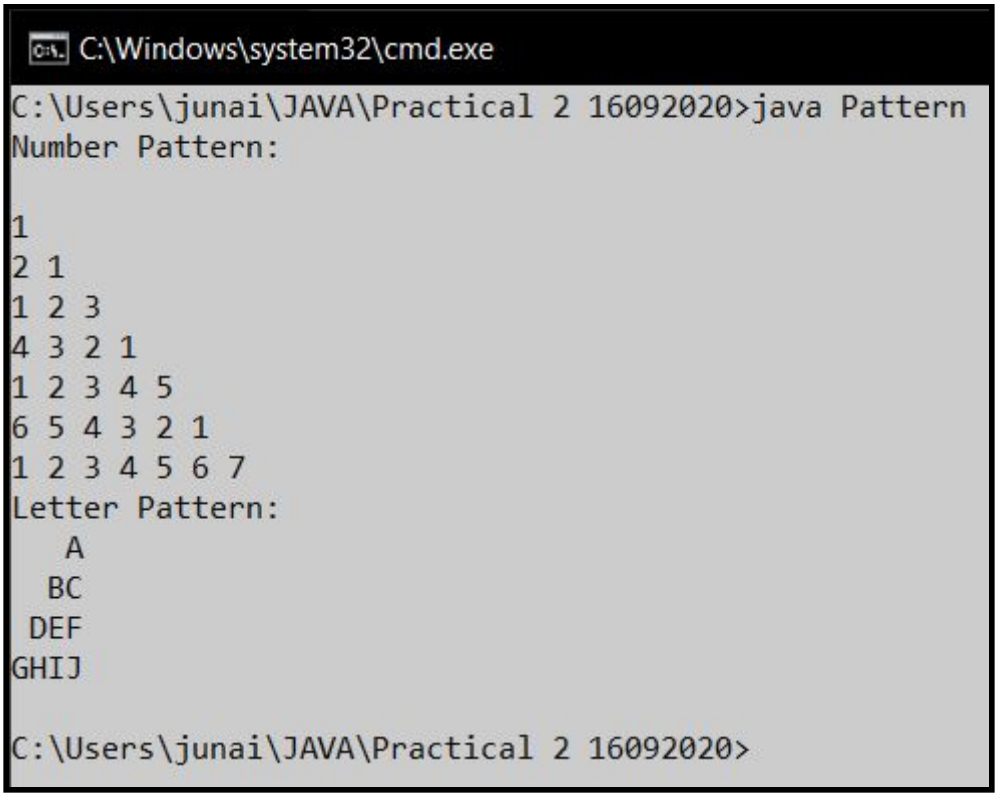
Q5. *WAP to display the following patterns:*

CODE:

```
class Pattern {
public static void main(String[] args) {
System.out.println("Number Pattern: ");
for (int i = 0; i <= 7; i++) {
if (i % 2 == 1) {
for (int j = 1; j <= i; j++) {
System.out.print(j + " ");
}
```

```
System.out.println("");
} else {
for (int k = i; k > 0; k--) {
System.out.print(k + " ");
}
System.out.println("");
}
}
int a = 0;
System.out.println("Letter Pattern:");
for (int h = 0; h < 4; h++) {
for (int g = 3; g > h; g--) {
System.out.print(" ");
}
for (int l = 0; l <= h; l++) {
System.out.print((char) (a + 65));
a++;
} System.out.println("");
}
}}
```

OUTPUT:



EXPERIMENT 3

AIM: To implement Arrays.

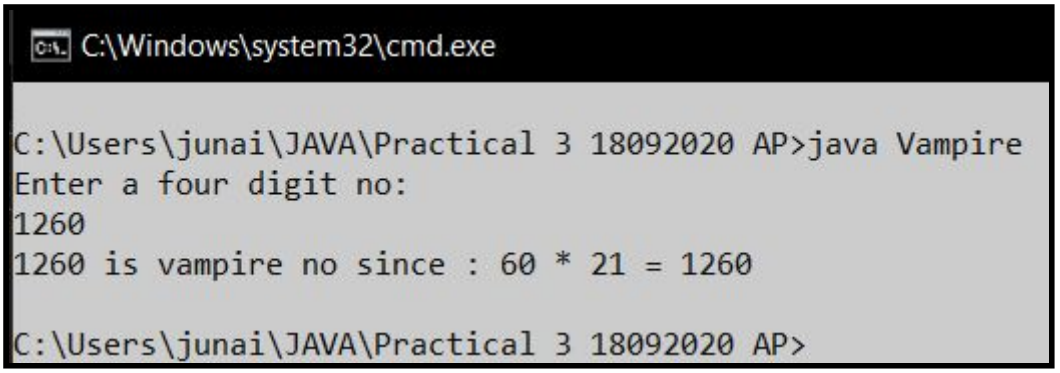
Q1. WAP to find whether the entered 4 digit number is vampire or not. Combination of digits from this number forms 2 digit number. When they are multiplied by each other we get the original number. (1260=21*60, 1395=15*93, 1530=30*51)

CODE:

```
import java.util.Scanner;

class Vampire {
	public static void main(String[] args) {
	Scanner in = new Scanner(System.in);
	String no = "";
	System.out.println("Enter a four digit no: ");
	while (no.length() != 4) {
		no = in.nextLine();
	}
	String a[] = no.split("");
	String c[][] = new String[12][2];
	int k = 0, i;
	for (i = 0; i < 4; i++) {
		for (int j = i + 1; j < 4; j++) {
			int m = -1, n = -1;
			int o[] = { 0, 1, 2, 3 };
			for (int p = 0; p < 4; p++) {
			if (o[p] != i && o[p] != j) {
				if (m == -1) {
					m = o[p];
				    continue;
				}
				if (n == -1) {
					n = o[p];
				}
			}
```

```
                    }
                    c[k][0] = a[i] + a[j];
                    c[k][1] = a[m] + a[n];
                    k++;
                    c[k][0] = a[i] + a[j];
                    c[k][1] = a[n] + a[m];
                    k++;
              }
        }
        int copy = Integer.parseInt(no);
        int flag = 0;
        for (i = 0; i < 12; i++) {
              if (Integer.parseInt(c[i][0]) * Integer.parseInt(c[i][1]) ==
copy) {
                    flag = 1;
                    break;
              }
        }
        if (flag == 1) {
              System.out.println(no + " is vampire no since : " + c[i][0]
+ " * " + c[i][1] + " = " + no);
        } else {
              System.out.println(no + " is not a vampire no");
        }
        }
}
```

OUTPUT:



Q2. WAP to display the following using irregular arrays

1

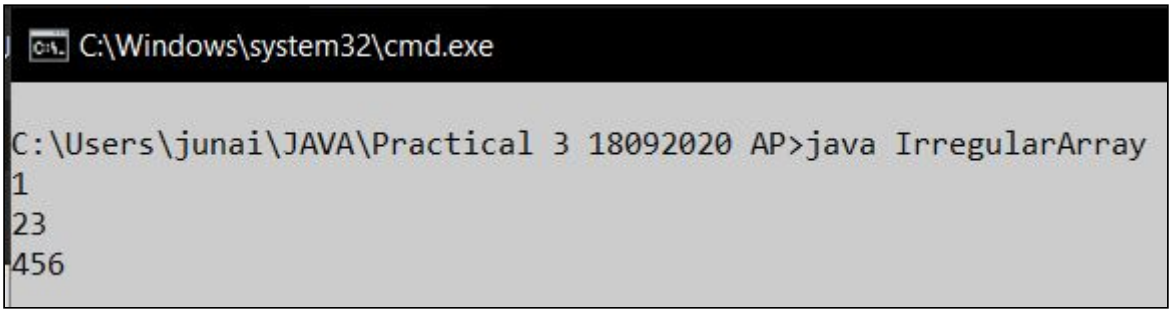
2 3

4 5 6

CODE:

```
class IrregularArray {
	public static void main(String[] args) {
	int a[][];
	a = new int[3][];
	a[0] = new int[1];
	a[1] = new int[2];
	a[2] = new int[3];
	int k = 1;
	for (int i = 0; i < a.length; i++) {
		for (int j = 0; j < a[i].length; j++) {
			a[i][j] = k++;
		}
	}
	for (int i = 0; i < a.length; i++) {
		for (int j = 0; j < a[i].length; j++) {
			System.out.print(a[i][j]);
		}
		System.out.println();
	}
	}
}
```

OUTPUT:



Q3.

Write a program that queries a user for the no.: of rows and columns representing students and their marks.
Reads data row by row and displays the data in tabular form along with the row totals, column totals and grand total
Hint : For the data 1, 3, 6, 7, 9, 8 the output is

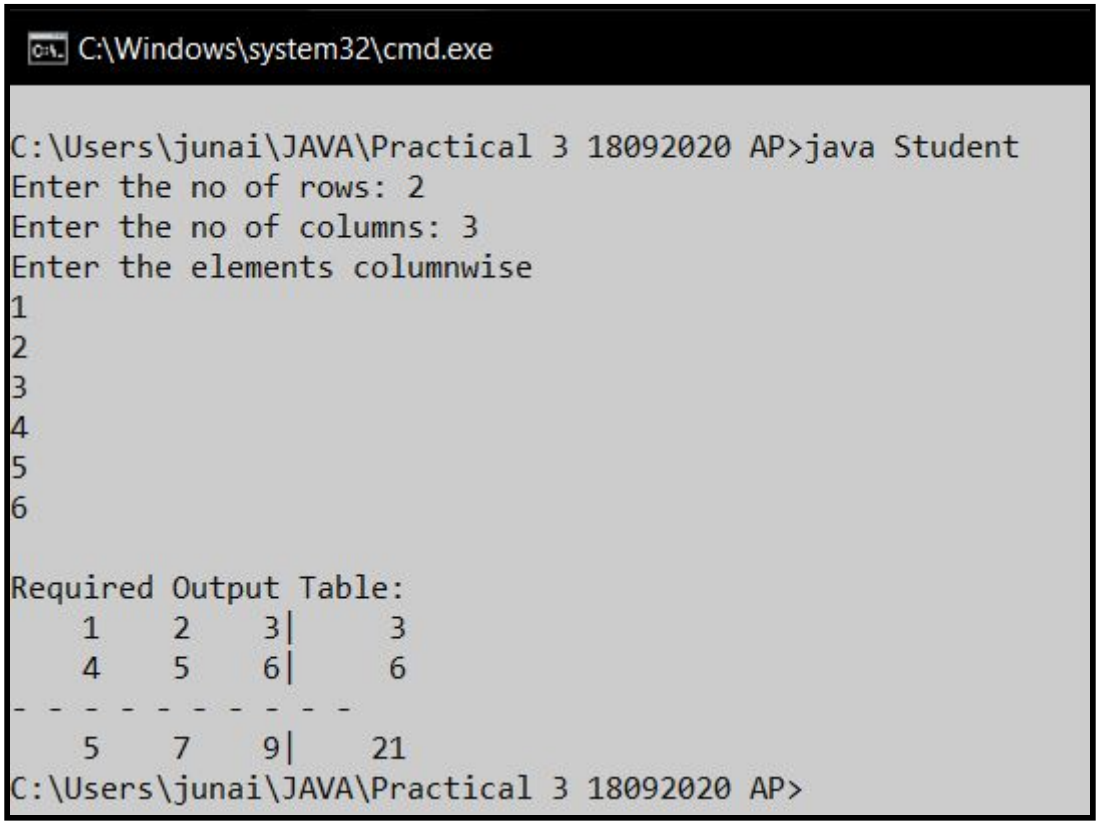
1	3	6	|	10
7	9	8	|	24
8	12	14	|	34

CODE:

```
import java.util.Scanner;

class Student {
	public static void main(String[] args) {
	int i, j;
	Scanner in = new Scanner(System.in);
	System.out.print("Enter the no of rows: ");
	int m = in.nextInt();
	System.out.print("Enter the no of columns: ");
	int n = in.nextInt();
	int table[][] = new int[m + 1][n + 1];
	System.out.println("Enter the elements columnwise");
	for (i = 0; i < m; i++) {
		for (j = 0; j < n; j++) {
			table[i][j] = in.nextInt();
		}
	}
	for (i = 0; i < m; i++) {
		int count = 0;
		for (j = 0; j < n; j++) {
			count += table[i][j];
		}
		table[i][n] = count;
	}
	for (j = 0; j < n + 1; j++) {
		int count = 0;
		for (i = 0; i < m; i++) {
			count += table[i][j];
		}
		table[m][j] = count;
```

```
        }
        System.out.println("\nRequired Output Table:");
        for (i = 0; i < m; i++) {
            for (j = 0; j < n; j++) {
                  System.out.printf("%5d", table[i][j]);
              }
              System.out.print("| ");
              System.out.printf("%5d\n", table[i][m]);
        }
        for (j = 0; j <= n + 1; j++) {
              System.out.print("- - ");
        }
          System.out.print("\n");
        for (j = 0; j < n; j++) {
              System.out.printf("%5d", table[m][j]);
        }
        System.out.print("| ");
        System.out.printf("%5d", table[m][j]);
    }
}
```

OUTPUT:



EXPERIMENT 4

AIM: To implement Vectors.

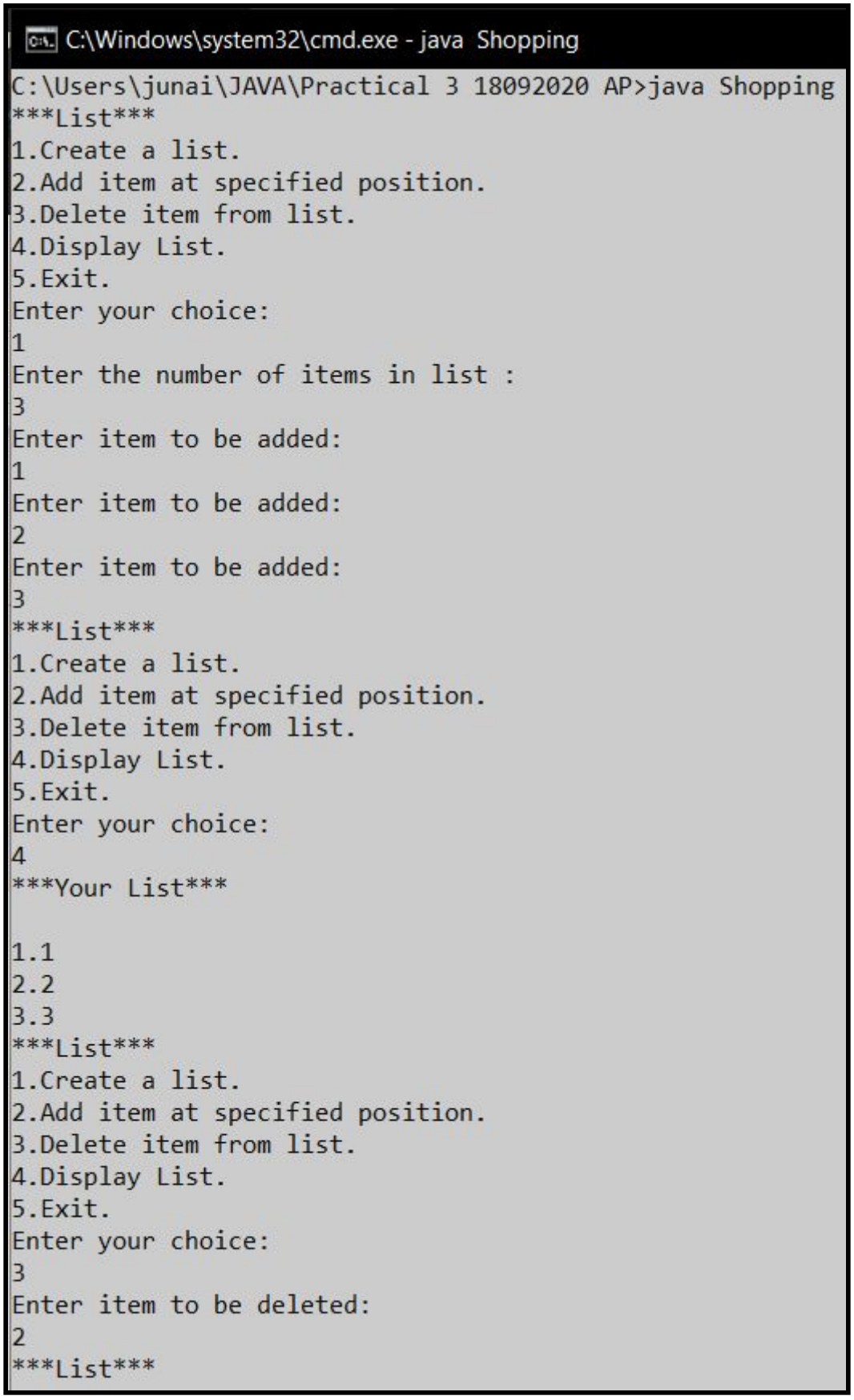
Q1. WAP that accepts a shopping list of items and performs the following operations: Add an item at a specified location, delete an item in the list, and print the contents of the vector.

CODE:

```
import java.util.Vector;
import java.util.Scanner;

class Shopping {
	public static void main(String[] args) {
	Vector list = new Vector(10, 10);
	Scanner in = new Scanner(System.in);
	String item;
	System.out.println(
			"***List***\n1.Create a list.\n2.Add item at specified
position.\n3.Delete item from list.\n4.Display List.\n5.Exit.");
	System.out.println("Enter your choice: ");
	int ch = in.nextInt();
	while (ch != 5) {
		switch (ch) {
			case 1:
			  System.out.println("Enter the number of items in
list : ");
			int len = in.nextInt();
			for (int j = 0; j < len; j++) {
				System.out.println("Enter item to be added: ");
			    item = in.next();
				list.addElement(item);
			}
			break;
			case 2:
			  System.out.println("Enter item to be added : ");
```

```
                  item = in.next();
                    System.out.println("Enter position where item to be
added: ");
                  int pos = in.nextInt();
                  list.insertElementAt(item, pos - 1);
                  break;
                  case 3:
                    System.out.println("Enter item to be deleted: ");
                  item = in.next();
                  list.removeElement(item);
                  break;
                  case 4:
                  System.out.println("***Your List***\n");
                  for (int i = 0; i < list.size(); i++) {
                        System.out.println((i + 1) + "." +
list.elementAt(i));
                  }
                  break;
            }
            System.out.println(
                  "***List***\n1.Create a list.\n2.Add item at specified
position.\n3.Delete item from list.\n4.Display List.\n5.Exit.");
            System.out.println("Enter your choice: ");
            ch = in.nextInt();
      }
      in.close();
      }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe - java Shopping
C:\Users\junai\JAVA\Practical 3 18092020 AP>java Shopping
List
1.Create a list.
2.Add item at specified position.
3.Delete item from list.
4.Display List.
5.Exit.
Enter your choice:
1
Enter the number of items in list :
3
Enter item to be added:
1
Enter item to be added:
2
Enter item to be added:
3
List
1.Create a list.
2.Add item at specified position.
3.Delete item from list.
4.Display List.
5.Exit.
Enter your choice:
4
Your List
1.1
2.2
3.3
List
1.Create a list.
2.Add item at specified position.
3.Delete item from list.
4.Display List.
5.Exit.
Enter your choice:
3
Enter item to be deleted:
2
List

```
1.Create a list.
2.Add item at specified position.
3.Delete item from list.
4.Display List.
5.Exit.
Enter your choice:
2
Enter item to be added :
5
Enter position where item to be added:
2
***List***
1.Create a list.
2.Add item at specified position.
3.Delete item from list.
4.Display List.
5.Exit.
Enter your choice:
4
***Your List***

1.1
2.5
3.3
***List***
```

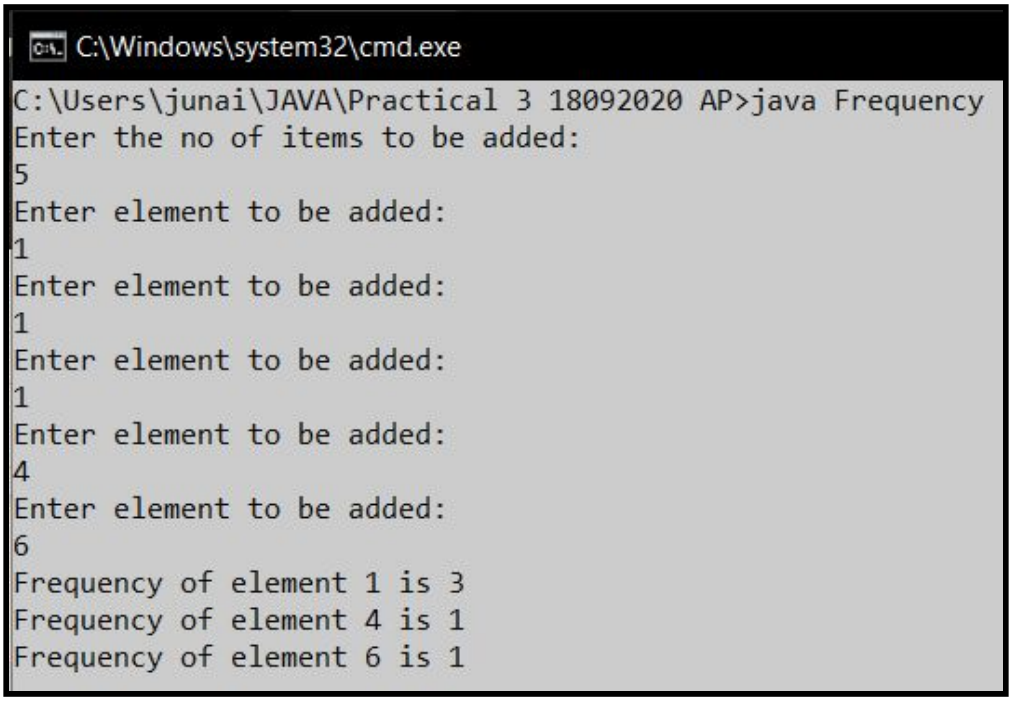
Q2. Write a java programs to find frequency of an element in the given Vector array.

CODE:

```java
import java.util.Vector;
import java.util.Scanner;

class Frequency {
	public static void main(String[] args) {
	Scanner in = new Scanner(System.in);
	int i;
	System.out.println("Enter the no of items to be added: ");
```

```
      int n = in.nextInt();
      Vector v = new Vector(n);
      int fr[] = new int[n];
      for (i = 0; i < n; i++) {
            System.out.println("Enter element to be added: ");
            String element = in.next();
            v.addElement(element);
      }
      int visited = -1;
      for (i = 0; i < n; i++) {
            if (fr[i] != visited) {
                  int count = 1;
                  for (int j = i + 1; j < n; j++) {
                  if
(((v.get(i)).toString()).equals(((v.get(j)).toString()))) {
                        count++;
                        fr[j] = visited;
                  }
                  }
                  fr[i] = count;
            }
      }
      for (i = 0; i < fr.length; i++) {
            if (fr[i] != visited)
                  System.out.println("Frequency of element " +
v.elementAt(i) + " is " + fr[i]);
      }
      }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 3 18092020 AP>java Frequency
Enter the no of items to be added:
5
Enter element to be added:
1
Enter element to be added:
1
Enter element to be added:
1
Enter element to be added:
4
Enter element to be added:
6
Frequency of element 1 is 3
Frequency of element 4 is 1
Frequency of element 6 is 1

EXPERIMENT 5

AIM: To implement Strings.

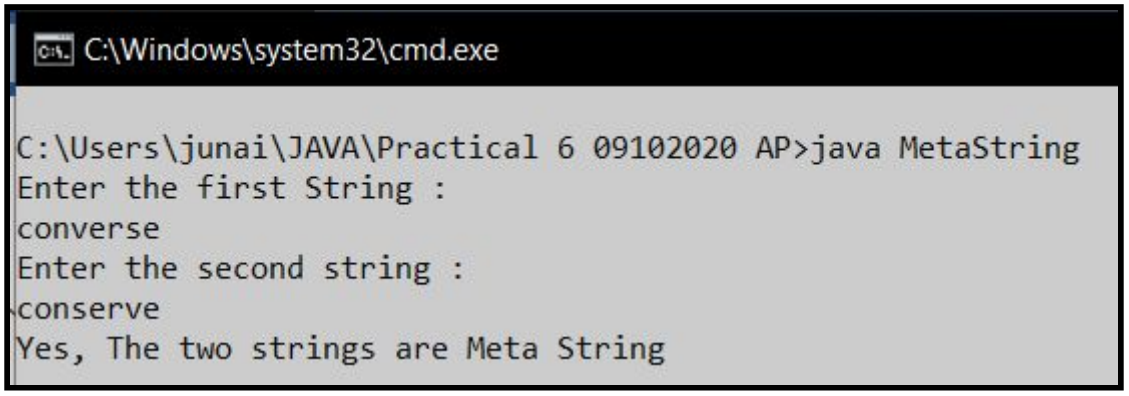
Q1. WAP to check if 2 strings are Meta strings or not. Meta strings are the strings which can be made equal by exactly one swap in any of the strings. Equal string are not considered here as Meta strings.

CODE:

```
import java.util.Scanner;

class MetaString {

	static boolean areMetaStrings(String str1, String str2) {
	int len1 = str1.length();
	    int len2 = str2.length();

	// Return false if both are not of equal length
	if (len1 != len2)
		return false;

	// To store indexes of previously mismatched characters
	int prev = -1, curr = -1;

	  int count = 0;
	for (int i = 0; i < len1; i++) {
		if (str1.charAt(i) != str2.charAt(i)) {
			// Count number of unmatched character
			count++;

		// If unmatched are greater than 2,
			// then return false
			if (count > 2)
			return false;
```

```
                    // Store both unmatched characters of both strings
                    prev = curr;
                    curr = i;
                }
            }


        return (count == 2 && str1.charAt(prev) == str2.charAt(curr) &&
str1.charAt(curr) == str2.charAt(prev));
        }


        public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first String : ");
        String string1 = sc.nextLine();
        System.out.println("Enter the second string : ");
        String string2 = sc.nextLine();

          System.out.println(areMetaStrings(string1, string2) ? "Yes, The
two strings are Meta String"
                    : "No, the two strings are not Meta Strings");

        }
}
```

OUTPUT:



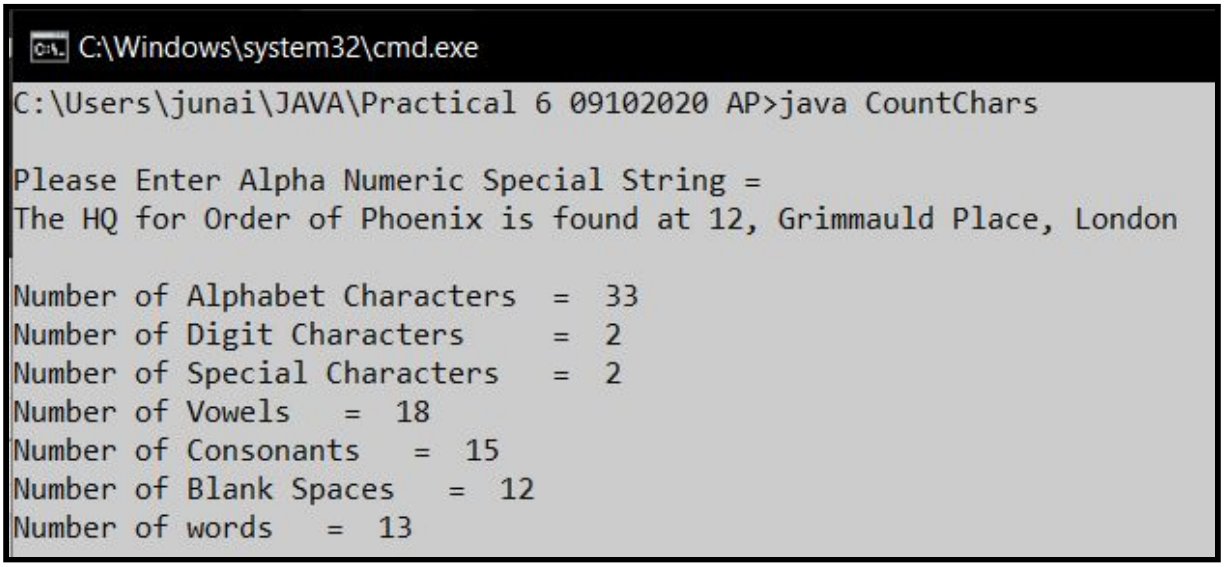
Q2. Write a java program to count number of alphabets, digits, special symbols, blank spaces and words from the given sentence. Also count number of vowels and consonants.

CODE:

```
import java.util.Scanner;

public class CountChars {

	public static void main(String[] args) {
	String enteredString;
	int i;
	int alph = 0;
	int digi = 0;
	int spl = 0;
	int vowel = 0;

	int blankSpace = 0;
	int words = 0;

	char ch;

	Scanner sc = new Scanner(System.in);
	System.out.println("\nPlease Enter Alpha Numeric Special String
=  ");
	enteredString = sc.nextLine();

	for (i = 0; i < enteredString.length(); i++) {
		ch = enteredString.charAt(i);
		if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch
== 'u') {
			vowel++;
		} else if (ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <=
'Z') {
			alph++;

		} else if (ch >= '0' && ch <= '9') {
			digi++;
		} else if (ch == ' ') {
			blankSpace++;
			words++;
```

```
            } else {
                  spl++;
            }
      }
      System.out.println("\nNumber of Alphabet Characters  =  " +
alph);
      System.out.println("Number of Digit Characters     =  " +
digi);
      System.out.println("Number of Special Characters   =  " + spl);
      System.out.println("Number of Vowels   =  " + vowel);
      System.out.println("Number of Consonants   =  " + (alph -
vowel));
      System.out.println("Number of Blank Spaces   =  " +
blankSpace);
      System.out.println("Number of words   =  " + (words + 1));
      }
}
```

OUTPUT:



EXPERIMENT 6

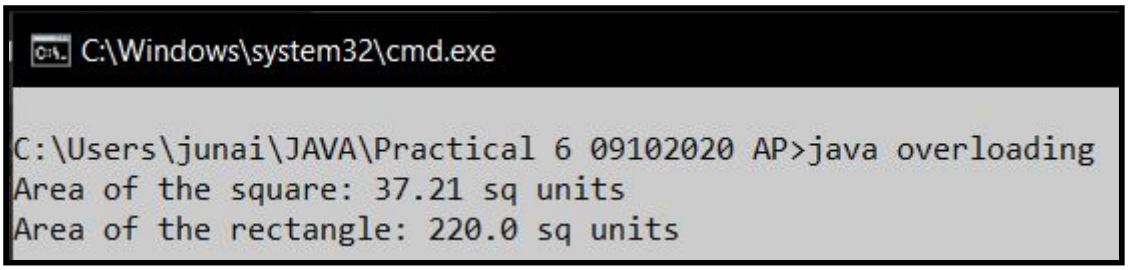
AIM: To implement Functions, recursive functions and overloading.

Q1. WAP to display area of square and rectangle using the concept of overloaded functions.

CODE:

```
class overloading {
    void calculateArea(float x) {
    System.out.println("Area of the square: " + x
* x + " sq units");
    }

    void calculateArea(float x, float y) {
    System.out.println("Area of the rectangle: "
+ x * y + " sq units");
    }

    public static void main(String args[]) {
    overloading obj = new overloading();

    obj.calculateArea(6.1f);

    obj.calculateArea(10, 22);

    }
}
```

OUTPUT:



Q2. Write menu driven program to implement recursive functions for following tasks.

a) To find GCD and LCM
b) To find X^Y
c) To print n Fibonacci numbers
d) To find reverse of number
e) To 1+2+3+4+…….+ (n-1)+n
f) Calculate sum of digits of a number

CODE:

```
import java.util.Scanner;

class Recursion {

    static int gcd(int a, int b) {

        if (a == 0)
            return b;
        if (b == 0)
            return a;

        if (a == b)
            return a;

        if (a > b)
            return gcd(a - b, b);
        return gcd(a, b - a);
    }

    static int lcm(int a, int b) {
        return (a / gcd(a, b)) * b;
```

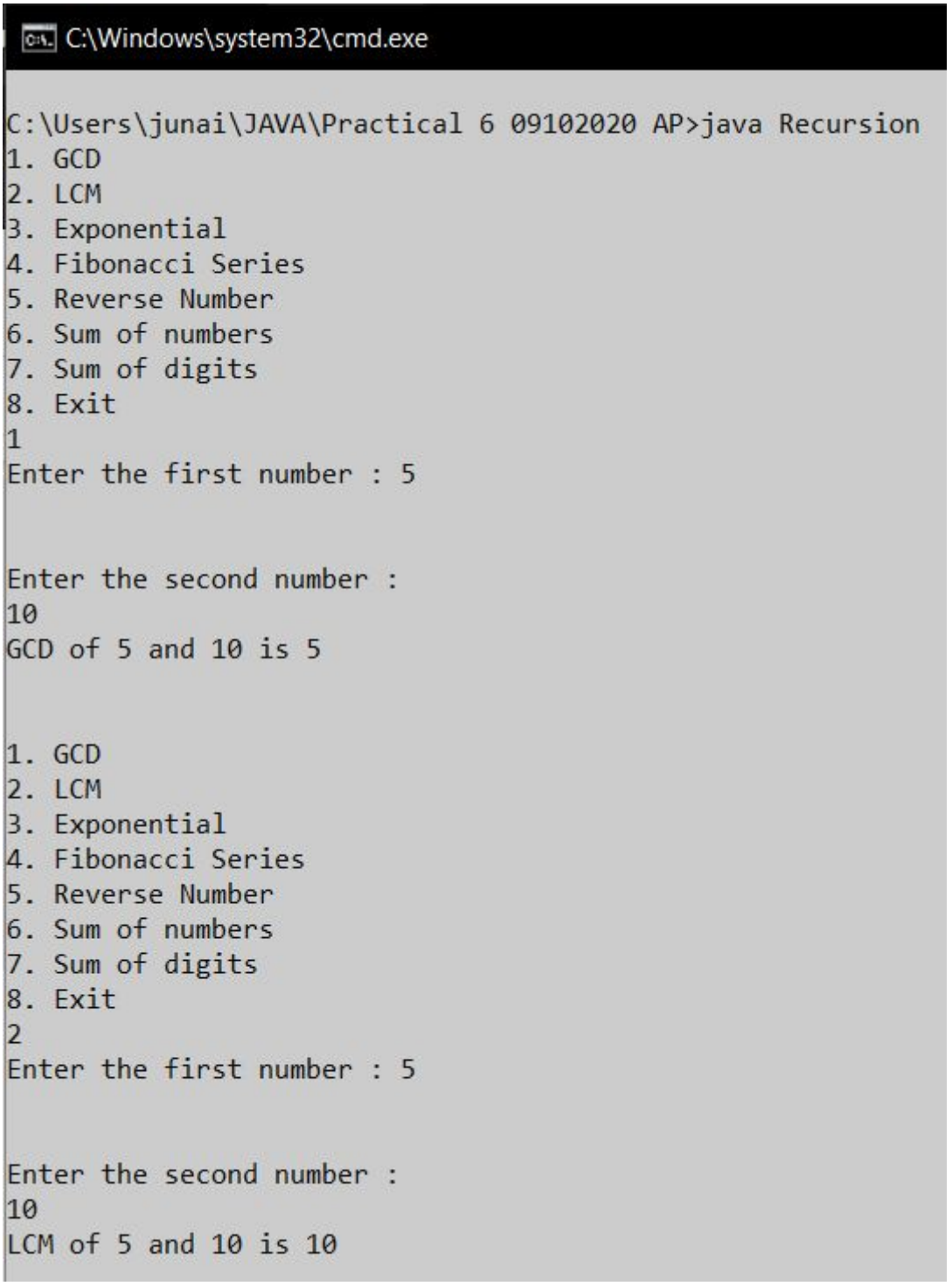
```
    }

    static int power(int x, int y) {
        if (y == 0)
            return 1;
        else if (y % 2 == 0)
            return power(x, y / 2) * power(x, y / 2);
        else
            return x * power(x, y / 2) * power(x, y / 2);
    }

    static void Fibonacci(int N) {
        int num1 = 0, num2 = 1;

        int counter = 0;

        while (counter < N) {

            System.out.print(num1 + " ");

            int num3 = num2 + num1;
            num1 = num2;
            num2 = num3;
            counter = counter + 1;
        }
    }

    static int reversDigits(int num) {
        int rev_num = 0;
        while (num > 0) {
            rev_num = rev_num * 10 + num % 10;
            num = num / 10;
        }
        return rev_num;
    }

    static int calculateSum(int n) {
        int sum = 0, i;
        for (i = 1; i <= n; i++) {
            sum += i;
        }
        return sum;
    }

    static int sum_of_digit(int n) {
        if (n == 0)
```

```
            return 0;
        return (n % 10 + sum_of_digit(n / 10));
    }

    // Driver method
    public static void main(String[] args) {
        int choice;
        Scanner sc = new Scanner(System.in);
        do {
            System.out.println(
                    "1. GCD\n2. LCM\n3. Exponential\n4. Fibonacci
Series\n5. Reverse Number\n6. Sum of numbers\n7. Sum of digits\n8.
Exit");
            choice = sc.nextInt();

            switch (choice) {
                case 1: {
                    System.out.print("Enter the first number : ");
                    int a = sc.nextInt();
                    System.out.println("\n");
                    System.out.println("Enter the second number : ");
                    int b = sc.nextInt();
                    System.out.println("GCD of " + a + " and " + b + "
is " + gcd(a, b));
                    System.out.println('\n');
                    break;
                }

                case 2: {
                    System.out.print("Enter the first number : ");
                    int a = sc.nextInt();
                    System.out.println("\n");
                    System.out.println("Enter the second number : ");
                    int b = sc.nextInt();
                    System.out.println("LCM of " + a + " and " + b + "
is " + lcm(a, b));
                    System.out.println('\n');
                    break;
                }

                case 3: {
                    System.out.print("Enter the base number : ");
                    int a = sc.nextInt();
                    System.out.println('\n');
                    System.out.println("Enter the power : ");
                    int b = sc.nextInt();
```

```
                    System.out.println(a + " raised to " + b + " gives "
+ power(a, b));
                    System.out.println('\n');
                    break;
                }

                case 4: {
                    System.out.print("Enter the count of fibonacci
numbers : ");
                    int a = sc.nextInt();
                    System.out.println('\n');
                    Fibonacci(a);
                    System.out.println('\n');
                    break;
                }

                case 5: {
                    System.out.print("Enter the number you want to
reverse : ");
                    int a = sc.nextInt();
                    System.out.println('\n');
                    System.out.println("Reverse of no. is " +
reversDigits(a));
                    System.out.println('\n');
                    break;
                }

                case 6: {
                    System.out.print("Enter the number you the sum upto
: ");
                    int a = sc.nextInt();
                    System.out.println('\n');
                    System.out.println("Sum of nnumbers is : " +
calculateSum(a));
                    System.out.println('\n');
                    break;
                }

                case 7: {
                    System.out.print("Enter the numberw whose digits sum
you want : ");
                    int a = sc.nextInt();
                    System.out.println('\n');
                    System.out.println("Sum of digits in " + a + " is "
+ sum_of_digit(a));
                    System.out.println('\n');
```

```
                    break;
                }

                case 8: {
                    break;
                }

                default: {
                    System.out.println("Invalid Choice");
                    System.out.println('\n');
                }
            }

        } while (choice != 8);

    }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 6 09102020 AP>java Recursion
1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
1
Enter the first number : 5
Enter the second number :
10
GCD of 5 and 10 is 5
1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
2
Enter the first number : 5
Enter the second number :
10
LCM of 5 and 10 is 10

```
C:\Windows\system32\cmd.exe

1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
3
Enter the base number : 2


Enter the power :
3
2 raised to 3 gives 8


1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
4
Enter the count of fibonacci numbers : 5


0 1 1 2 3

1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
5
Enter the number you want to reverse : 245


Reverse of no. is 542
```

```
C:\Windows\system32\cmd.exe


1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
6
Enter the number you the sum upto : 9


Sum of nnumbers is : 45


1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
7
Enter the numberw whose digits sum you want : 6543


Sum of digits in 6543 is 18


1. GCD
2. LCM
3. Exponential
4. Fibonacci Series
5. Reverse Number
6. Sum of numbers
7. Sum of digits
8. Exit
8
```

EXPERIMENT 7

AIM: To implement Array of Objects.

Q1. WOOP to arrange the names of students in descending order of their total marks, input data consists of student details such as names, ID.no, marks of maths, physics, chemistry. (Use array of objects).

CODE:

```
import java.util.Scanner;

public class ArrayofObjects {
	public static void main(String[] args) {
	Scanner ob = new Scanner(System.in);
	int n;
	int i;
	int s_id;
	String name;
	long math_mark;
	long chem_mark;
	long phy_mark;
	System.out.println("Enter number of Students");
	n = ob.nextInt();
	Student[] objArr = new Student[n];
	System.out.println("Enter The Student Details");
	for (i = 0; i < n; i++) {
		ob.nextLine();
		System.out.print("Enter Student Name: ");
		name = ob.nextLine();
		System.out.println();
		System.out.print("Enter Student ID: ");
		s_id = ob.nextInt();
		System.out.println();
		 System.out.print("Enter Math Marks of Student: ");
		math_mark = ob.nextInt();
		System.out.println();
		System.out.print("Enter Chem Marks of Student: ");
```

```
            chem_mark = ob.nextInt();
            System.out.println();
            System.out.print("Enter Physics Marks of Student: ");
            phy_mark = ob.nextInt();
            System.out.println();
            objArr[i] = new Student(name, s_id, phy_mark, chem_mark,
math_mark);
            System.out.println("=============== Student Details Have
Successfully Been Entered ==================");

      }
      ob.close();
        System.out.println("================== Student Details:
==========================");
        for (i = 0; i < n; i++) {
            System.out.println("\nName = " + objArr[i].getName() + "
Student ID: " + objArr[i].getID());
      }
      sort(objArr);
      System.out.println();
        System.out.println("==================== Sorted Students!
======================");
      for (i = 0; i < n; i++) {
            System.out.println("\nName = " + objArr[i].getName() + "
Student ID: " + objArr[i].getID());
      }
      }

      static void sort(Student[] objArr) {
      for (int i = 0; i < objArr.length - 1; i++)
            for (int j = 0; j <= objArr.length - i - 2; j++)
                  if (objArr[j].getTotalMarks() < objArr[j +
1].getTotalMarks()) {
                  Student temp = objArr[j];
                  objArr[j] = objArr[j + 1];
                  objArr[j + 1] = temp;
                  }
      }

}

class Student {

      // attributes | class instance variables
      private String name;
      private int s_ID;
```

```
	private long phy_marks, chem_marks, math_marks;
	private long total_marks;

	// constructor
	Student(String name, int s_ID, long phy_marks, long chem_marks,
long math_marks) {
	this.name = name;
	this.math_marks = math_marks;
	this.phy_marks = phy_marks;
	this.chem_marks = chem_marks;
	this.s_ID = s_ID;
	this.total_marks = this.phy_marks + this.math_marks +
this.chem_marks;
	}

	public String getName() {
	return this.name;
	}

	public int getID() {
	return this.s_ID;
	}

	public long getTotalMarks() {
	return this.total_marks;
	}

	public long getMathMarks() {
	return this.math_marks;
	}

	public long getPhyMarks() {
	return this.phy_marks;
	}

	public long getChemMarks() {
	return this.chem_marks;
	}

}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 7 16102020 AP>java ArrayofObjects
Enter number of Students
2
Enter The Student Details
Enter Student Name: FIrst

Enter Student ID: 1

Enter Math Marks of Student: 99

Enter Chem Marks of Student: 99

Enter Physics Marks of Student: 99

=============== Student Details Have Succussfully Been Entered ===================
Enter Student Name: Second

Enter Student ID: 2

Enter Math Marks of Student: 100

Enter Chem Marks of Student: 100

Enter Physics Marks of Student: 100

=============== Student Details Have Succussfully Been Entered ===================
=================== Student Details: ============================

Name = FIrst Student ID: 1

Name = Second Student ID: 2

==================== Sorted Students! ======================

Name = Second Student ID: 2

Name = FIrst Student ID: 1

C:\Users\junai\JAVA\Practical 7 16102020 AP>
```

EXPERIMENT 8

AIM: To implement Constructors and overloading.

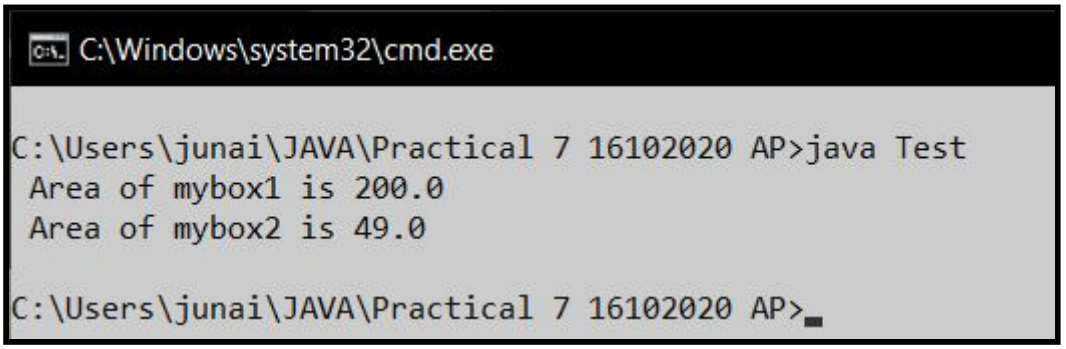
Q1. WAP find area of square and rectangle using overloaded constructor.

CODE:

```
class Box {
      double width, height;

      Box(double w, double h) {
      width = w;
      height = h;

      }

      // constructor used when square is created
      Box(double len) {
      width = height = len;
      }

      double area() {
      return width * height;
      }
}

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

      Box mybox1 = new Box(10, 20);
      Box mybox2 = new Box(7);

      double area;

      // get area of first box
      area = mybox1.area();
      System.out.println(" Area of mybox1 is " + area);

      // get area of second box
      area = mybox2.area();
```

```
        System.out.println(" Area of mybox2 is " + area);

        }
}
```

OUTPUT:



Q2. Create Rectangle and Cube class that encapsulates the properties of a rectangle and cube i.e. Rectangle has default and parameterized constructor and area() method. Cube has default and parameterized constructor and volume() method. They share no ancestor other than Object.

Implement a class Size with size() method. This method accepts a single reference argument z. If z refers to a Rectangle then size(z) returns its area and if z is a reference of Cube, then z returns its volume. If z refers to an object of any other class, then size(z) returns -1. Use the main method in Size class to call size(z) method.

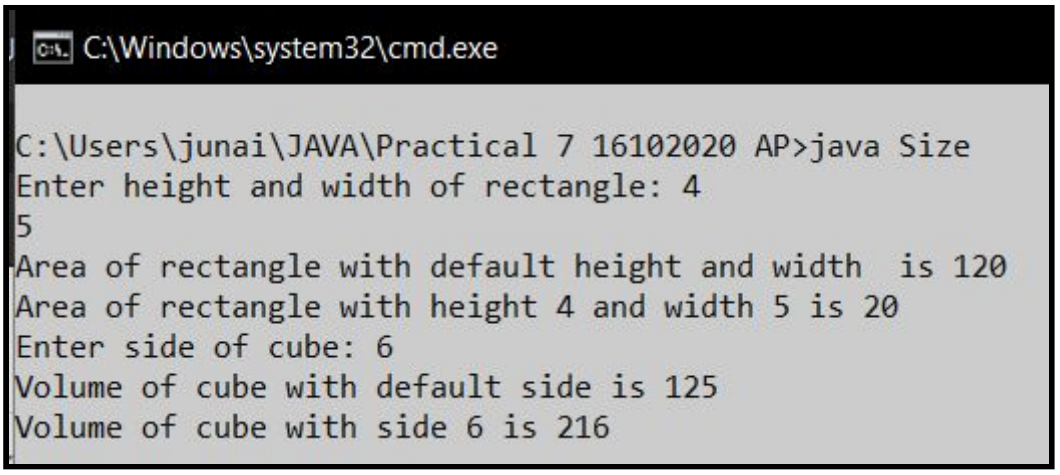
CODE:

```
import java.util.*;

class Rectangle {
    int x, y;

    public Rectangle() {
    x = 10; // Default length

    y = 12; // Default Width
    }

    public Rectangle(int height, int width) {
    x = height;
    y = width;
    }

    public int area() {
    return x * y;
    }

}

class Cube {
    int x;

    public Cube() {
    x = 5; // Default side of cube
    }

    public Cube(int side) {
    x = side;
```

```
    }

    public int volume() {
    return x * x * x;
    }

}

class Size {
    public static int size(Object obj) {
    if (obj instanceof Rectangle) {
        return ((Rectangle) obj).area();
    } else if (obj instanceof Cube) {
        return ((Cube) obj).volume();
    } else {
        return -1;
    }
    }

    public static void main(String args[]) {
    Scanner in = new Scanner(System.in);
    int x, y;

    System.out.print("Enter height and width of
rectangle: ");
    x = in.nextInt();
    y = in.nextInt();
    Rectangle defrec = new Rectangle();
    System.out.println("Area of rectangle with
default height and width  is " + size(defrec));
    Rectangle rec = new Rectangle(x, y);
    System.out.println("Area of rectangle with
```

```
height " + x + " and width " + y + " is " +
size(rec));

    System.out.print("Enter side of cube: ");
    x = in.nextInt();
    Cube defcb = new Cube();
    System.out.println("Volume of cube with
default side is " + size(defcb));
    Cube cb = new Cube(x);
    System.out.println("Volume of cube with side
" + x + " is " + size(cb));
    }
}
```

OUTPUT:



EXPERIMENT 9

AIM: To implement Abstract Classes

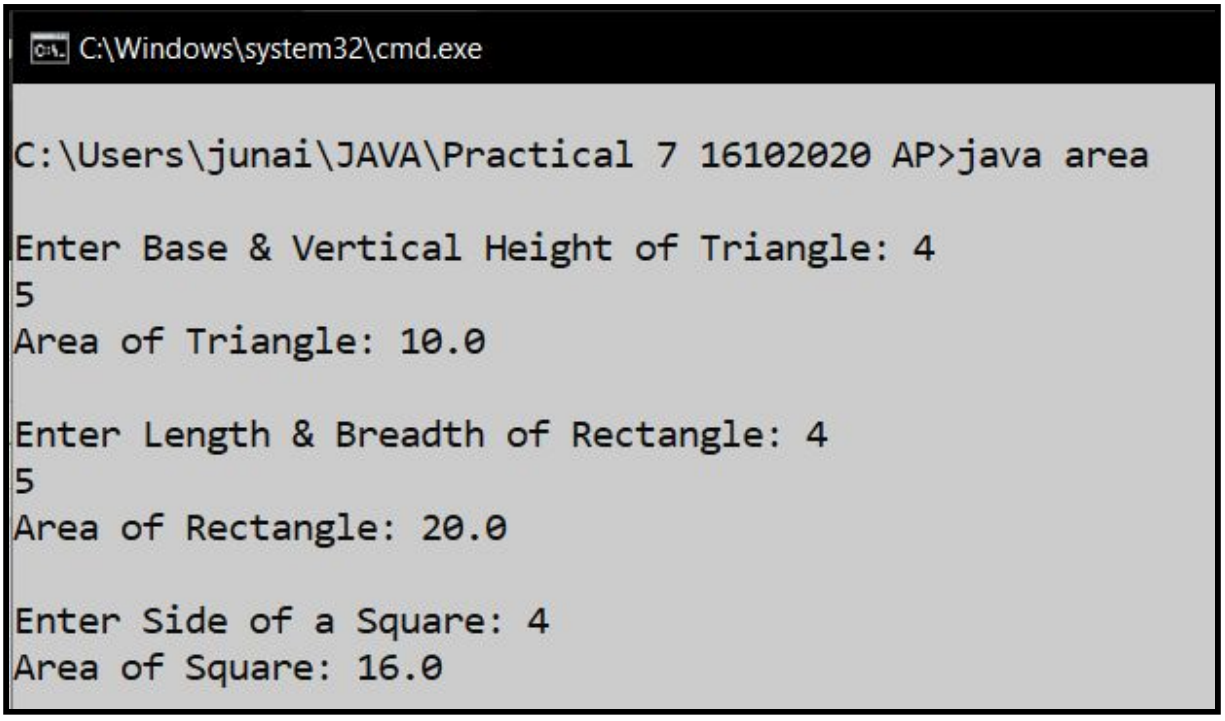
Q1. Write an abstract class program to calculate the area of the circle, rectangle and triangle.

CODE:

```
import java.util.Scanner;

abstract class calcArea {
	abstract void findTriangle(double b, double h);

	abstract void findRectangle(double l, double b);

	abstract void findSquare(double s);

}

class findArea extends calcArea {

	void findTriangle(double b, double h) {
	double area = (b * h) / 2;
	System.out.println("Area of Triangle: " + area);
	}

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

	void findSquare(double s) {
	double area = s * s;
	System.out.println("Area of Square: " + area);
	}

}

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

```
        double l, b, h, s;
        findArea area = new findArea();
        Scanner get = new Scanner(System.in);

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

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

        System.out.print("\nEnter Side of a Square: ");
        s = get.nextDouble();
        area.findSquare(s);

        }
}
```

OUTPUT:



EXPERIMENT 10

AIM: To implement Inheritance, interfaces and method overriding.

Q1. WAP to implement three classes namely Student, Test and Result. Student class has members as roll no, Test class has members as sem1_marks and sem2_marks and Result class has members as total. Create an interface named sports that has a member score (). Derive Test class from Student and Result class has multiple inheritances from Test and Sports. Total is a formula based on sem1_marks, sem2_mark and score.

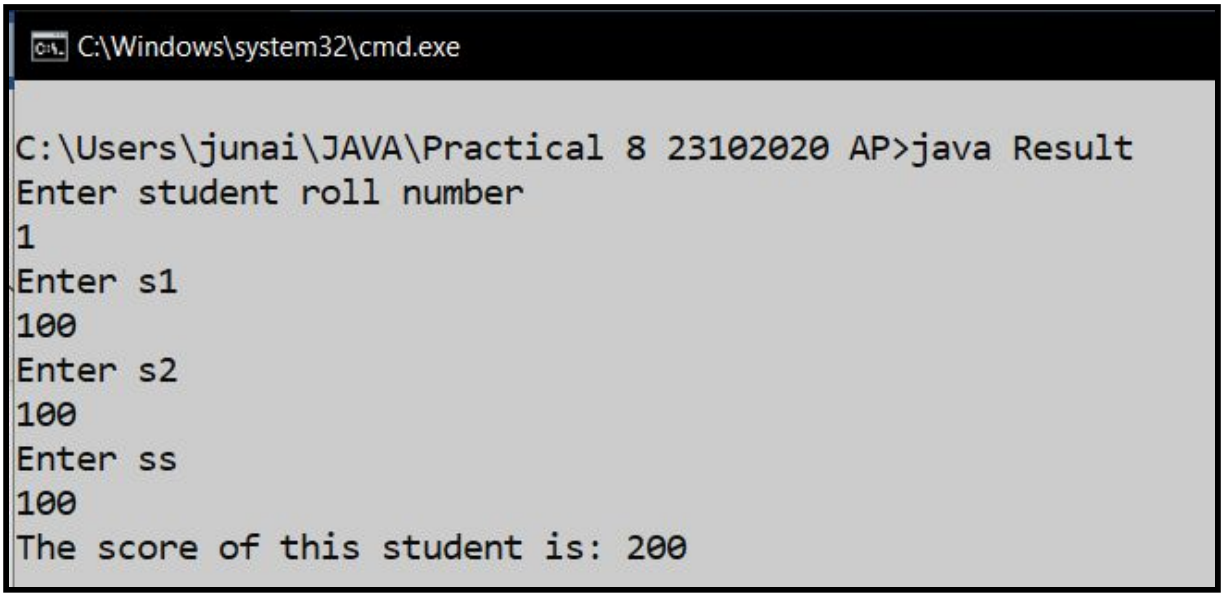
CODE:

```
import java.util.Scanner;

class Student {
      int rollNo;

      Student(int rollNo) {

      this.rollNo = rollNo;
      }

      void setRollno(int rollNo) {
      this.rollNo = rollNo;
      }

      int getRollNo() {
      return this.rollNo;
      }

}
```

```
class Test extends Student {
      int sem1Marks;
      int sem2Marks;

      Test(int sem1Marks, int sem2Marks, int rollNo) {
      super(rollNo);

      this.sem1Marks = sem1Marks;
      this.sem2Marks = sem2Marks;
      }

      void setSem1Marks(int sem1) {
      this.sem1Marks = sem1;
      }

      int getSem1Marks() {
      return this.sem1Marks;
      }

      void setSem2Marks(int sem2Marks) {
      this.sem2Marks = sem2Marks;
      }

      int getRollNo() {
      return this.sem2Marks;
      }
}

class Result extends Test implements Sports {
      int total;
      int sportsScore;

      Scanner ob = new Scanner(System.in);

      Result(int sem1Marks, int sem2Marks, int rollNo, int
sportScore) {

      super(sem1Marks, sem2Marks, rollNo);

      this.sportsScore = sportScore;
      }

      public void score() {
      this.total = (sem1Marks + sem2Marks) / 2 + sportsScore;
```

```
      System.out.println("The score of this student is: " +
this.total);
      }

      public static void main(String[] args) {
      int r, s1, s2, s;
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter student roll number");
      r = sc.nextInt();
        System.out.println("Enter s1");
      s1 = sc.nextInt();
      System.out.println("Enter s2");
      s2 = sc.nextInt();
      System.out.println("Enter ss");
      s = sc.nextInt();
      Result res = new Result(s1, s2, r, s);
      res.score();
      }

}

interface Sports {
      void score();
}
```

OUTPUT:



EXPERIMENT 11

AIM: To implement Package

Q1. WAP to create a user defined package & import the package in another program.

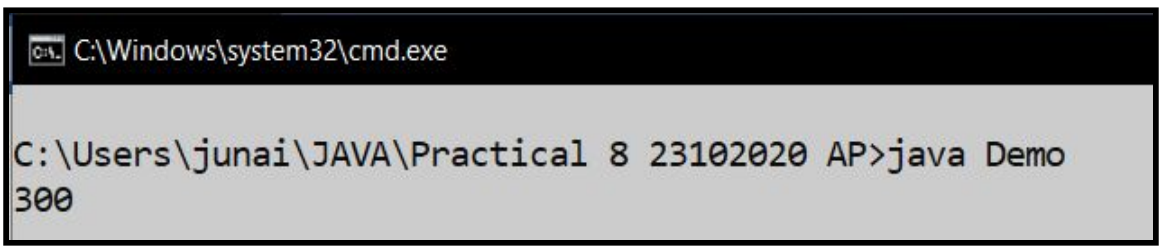
CODE:

PACKAGE (..\letmeadd\Calculator.java):

```java
package letmeadd;

public class Calculator {
   public int add(int a, int b){
            return a+b;
   }
   public static void main(String args[]){
            Calculator obj = new Calculator();
            System.out.println(obj.add(10, 20));
   }
}
```

```java
import letmeadd.Calculator;

public class Demo {
    public static void main(String args[]) {
        Calculator obj = new Calculator();
        System.out.println(obj.add(100, 200));
    }
}
```

OUTPUT:



EXPERIMENT 12

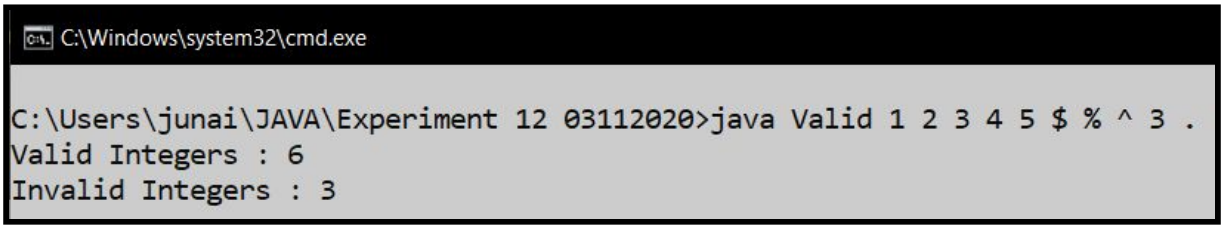
AIM: To implement exceptions in Java.

Q1. Write a Java Program to input the data through the command Line and Find out total valid and in-valid integers. (Hint: use exception handling).

CODE:

```
class Valid
{
    public static void main(String[] args)
      {
        int valid=0,invalid=0,temp;
        for(int i=0;i<args.length;i++)
        {
            try
            {
                temp=Integer.parseInt(args[i]);
                valid ++;
            }
            catch(NumberFormatException e)
            {
                invalid++;
            }
        }
        System.out.println("Valid Integers : " + valid + "\nInvalid Integers
: " + invalid);
    }
}
```

OUTPUT:



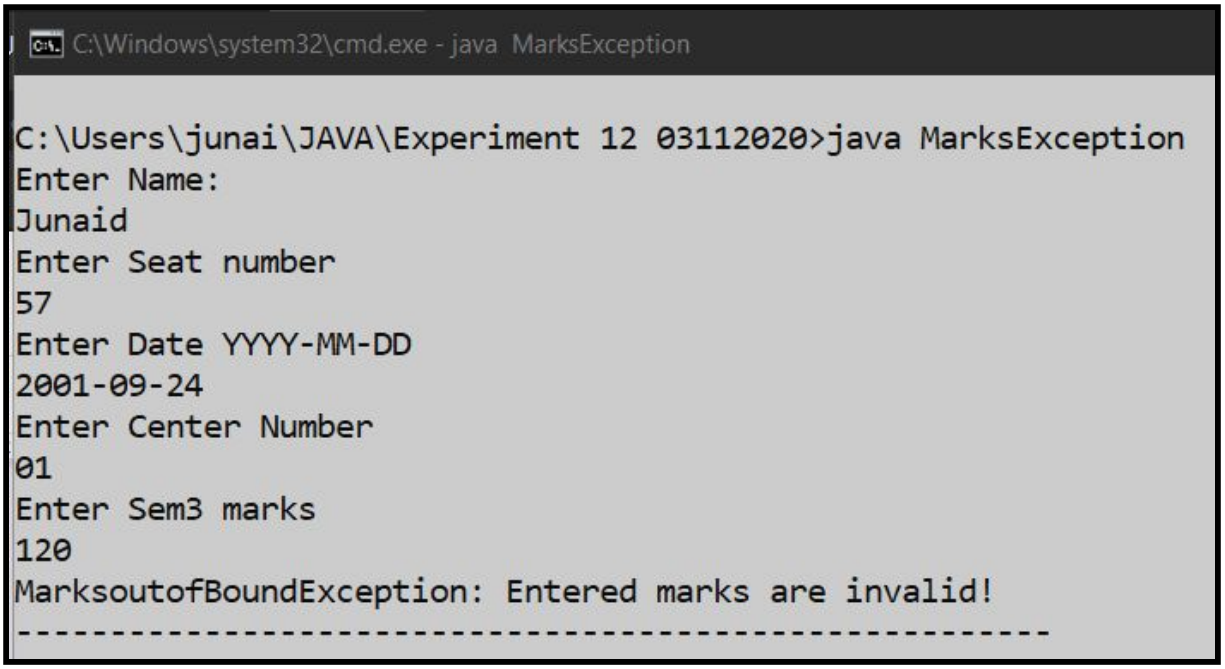
Q2. Write a Java Program to calculate the Result. Result should consist of name, seat no, date, center number and marks of semester three exam. Create a User Defined Exception class MarksOutOfBoundsException, If Entered marks of any subject is greater than 100 or less than 0, and then program should create a user defined Exception of type MarksOutOfBoundsException and must have a provision to handle it.

CODE:

```
import java.util.*;
class MarksoutofBoundException extends Exception
{
	MarksoutofBoundException(String s)
	{
	super(s);
	}
}
class Result
{
	String name, date;
	int seatno, centerno, sem3_marks;
	Result(String a, int b, String c, int d, int e)
	{
	name = a;
	seatno = b;
	date = c;
	centerno = d;
	sem3_marks = e;
	}
}
class MarksException
{
	static void verify(int marks) throws MarksoutofBoundException
	{
	if(marks<0 | marks>100)
	{
		throw new MarksoutofBoundException("Entered marks are
invalid!");
```

```
	}
	else
	{
		System.out.println("Valid Marks");
	}
	}
	public static void main(String[] args)
	{
	Scanner in = new Scanner(System.in);
	while(true){
	System.out.println("Enter Name:");
	String name = in.nextLine();

	System.out.println("Enter Seat number");
	int seatno = in.nextInt();
	in.nextLine();

	System.out.println("Enter Date YYYY-MM-DD");
	String date = in.nextLine();

	System.out.println("Enter Center Number");
	int centerno = (in.nextInt());

	System.out.println("Enter Sem3 marks");
	int sem3_marks = (in.nextInt());
	in.nextLine();

	Result res1 = new Result(name, seatno, date, centerno,
sem3_marks);
	try
	{
		verify(res1.sem3_marks);
	}
	catch(Exception e)
	{
		System.out.println(e);
	}

System.out.println("-----------------------------------------------------
---\n\n");
	}

	}
}
```

OUTPUT :


C:\Windows\system32\cmd.exe - java MarksException
C:\Users\junai\JAVA\Experiment 12 03112020>java MarksException
Enter Name:
Junaid
Enter Seat number
57
Enter Date YYYY-MM-DD
2001-09-24
Enter Center Number
01
Enter Sem3 marks
120
MarksoutofBoundException: Entered marks are invalid!

EXPERIMENT 13

AIM: To implement Multithreading.

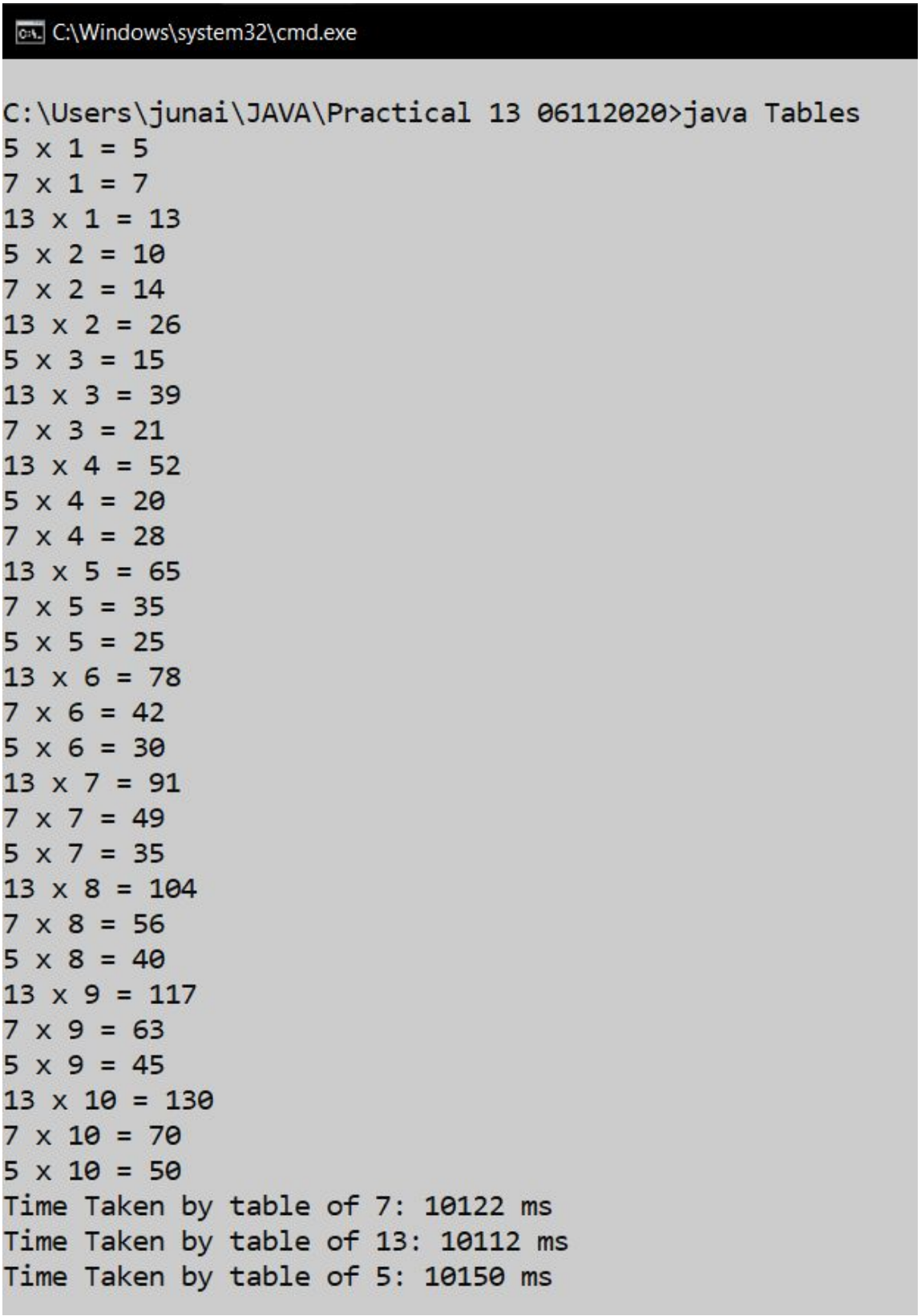
Q1. Write a java program to print Table of Five, Seven and Thirteen using Multithreading (Use Thread class for the implementation). Also print the total time taken by each thread for the execution.

CODE:

```
public class Tables {
    public static void main(String[] args) {
        Five f = new Five();
        Seven s = new Seven();
        Thirteen t = new Thirteen();
        f.start();
        s.start();
        t.start();
    }
}

class Seven extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
        for (int i = 1; i <= 10; i++) {
            System.out.println("7 x " + i + " = " + (7 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 7: " + (t2 - t1) + " ms "
);
    }
}

class Five extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
```

```
        for (int i = 1; i <= 10; i++) {
            System.out.println("5 x " + i + " = " + (5 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 5: " + (t2 - t1) + " ms "
);
    }
}

class Thirteen extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
        for (int i = 1; i <= 10; i++) {
            System.out.println("13 x " + i + " = " + (13 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 13: " + (t2 - t1) + " ms
" );
    }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 13 06112020>java Tables
5 x 1 = 5
7 x 1 = 7
13 x 1 = 13
5 x 2 = 10
7 x 2 = 14
13 x 2 = 26
5 x 3 = 15
13 x 3 = 39
7 x 3 = 21
13 x 4 = 52
5 x 4 = 20
7 x 4 = 28
13 x 5 = 65
7 x 5 = 35
5 x 5 = 25
13 x 6 = 78
7 x 6 = 42
5 x 6 = 30
13 x 7 = 91
7 x 7 = 49
5 x 7 = 35
13 x 8 = 104
7 x 8 = 56
5 x 8 = 40
13 x 9 = 117
7 x 9 = 63
5 x 9 = 45
13 x 10 = 130
7 x 10 = 70
5 x 10 = 50
Time Taken by table of 7: 10122 ms
Time Taken by table of 13: 10112 ms
Time Taken by table of 5: 10150 ms

Q2. Write java program to implement the concept of Thread Synchronization.

CODE:

```
public class Synchronization {
    public static void main(String[] args) {
        Flight f = new Flight(2);
        Thread t1 = new Thread(f);
        t1.setName("Tom");
        Thread t2 = new Thread(f);
        t2.setName("Harry");
        Thread t3 = new Thread(f);
        t3.setName("Dick");
        t1.start();
        t2.start();
        t3.start();
    }
}

class Flight extends Thread {
    int vacant = 5, required;

    Flight(int required) {
        this.required = required;
    }

    public synchronized void run() {
        if (vacant >= required) {
            System.out.println(required + " Tickets Have Been Booked For: "
+ Thread.currentThread().getName() + "!");
            try {
                Thread.sleep(100);
            } catch (Exception e) {
            }
            vacant -= required;
        } else {
            System.out.println("Sorry " + Thread.currentThread().getName() +
", But The Flight is Booked! (Only "
                    + vacant + " Seat(s) Available)");
        }
    }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
C:\Users\junai\JAVA\Practical 13 06112020>java Synchronization
2 Tickets Have Been Booked For: Tom!
2 Tickets Have Been Booked For: Dick!
Sorry Harry, But The Flight is Booked! (Only 1 Seat(s) Available)

EXPERIMENT 14

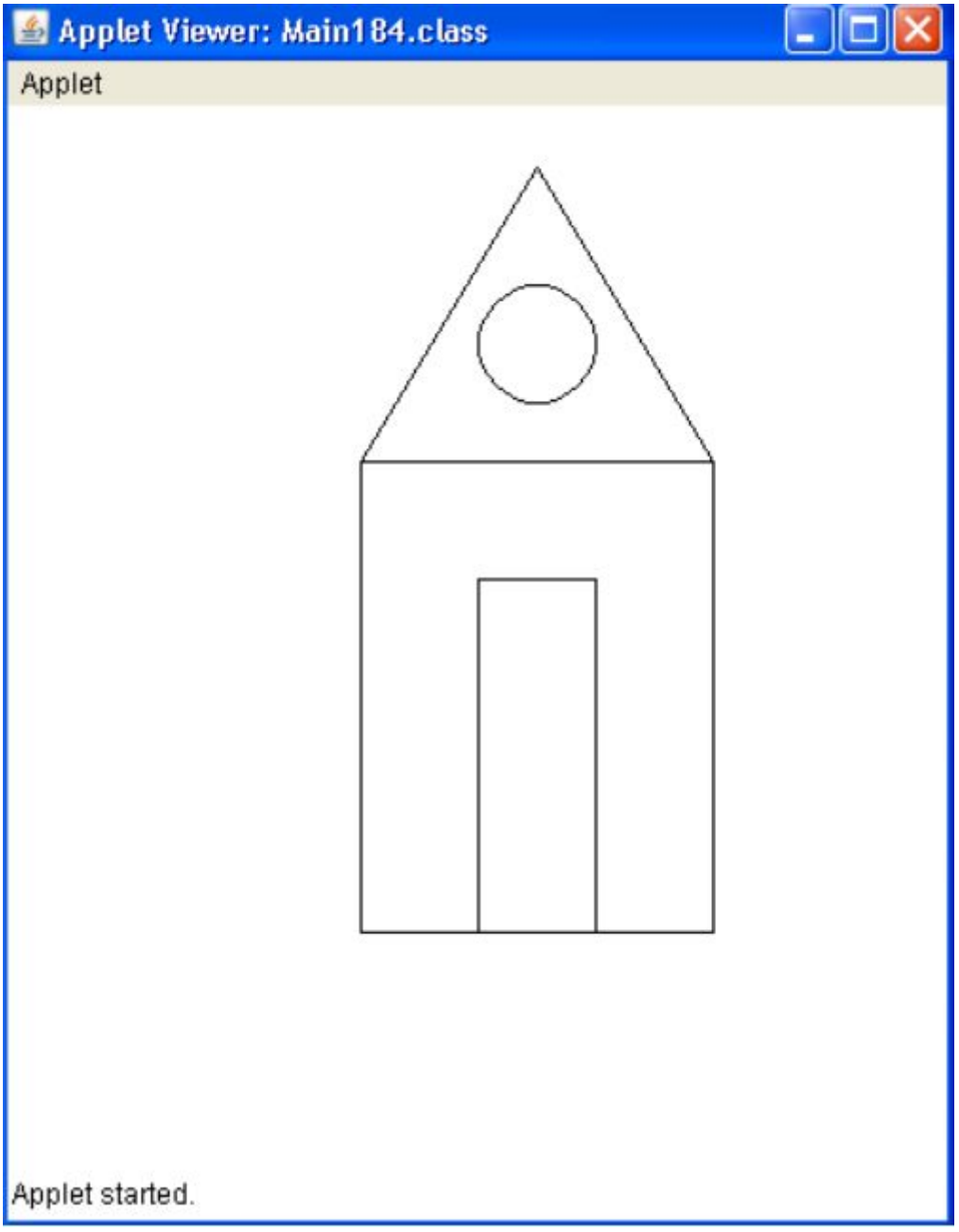
AIM: Designing Graphical User Interfaces in Java using AWT and Event handling.

Q1. Write java program to draw the house on an applet

CODE:

```
import java.awt.*;
import java.applet.*;
public class Main184 extends Applet
{
/*
<applet code="Main184.class" width=400 height=450></applet>
*/
public void paint(Graphics gp)
{ int [] x = {150, 300, 225};
int [] y = {150, 150, 25};
gp.drawRect(150, 150, 150, 200); //House
gp.drawRect(200, 200, 50, 150); // Door
gp.drawOval(200, 75, 50, 50); // Skylight
gp.drawPolygon(x, y, 3); // Roof
}
}
```

OUTPUT:



Q2. Write java program to create an advertisement banner on an applet using multithreading

CODE:

```
import java.applet.*;
import java.awt.*;
public class Banner extends Applet implements Runnable
{
String text = " Sample Banner ";
Thread t;
//Initialize the applet
public void init()
{
setBackground(Color.white);
}
//Function to start the thread
public void start()
{
t = new Thread(this);
t.start();
}
//Function to execute the thread
public void run()
{
while(true)
{
try
{
repaint();
//Delay each thread by 1000ms or 1 seconds
Thread.sleep(1000);
//Shift the first character of banner text to the last postion
text = text.substring(1) + text.charAt(0);
}
catch(Exception e)
{
}
}
}
//Function to draw text
public void paint(Graphics g)
{
g.setFont(new Font("TimesRoman",Font.BOLD,15));
g.drawString(text,200,200);
}
}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class MyFrame extends JFrame implements ActionListener {

private static final long serialVersionUID = 1L;
// Components of the Form
private Container c;
private JLabel title;
private JLabel name;
private JTextField tname;
private JLabel mno;
private JTextField tmno;
private JLabel gender;
private JRadioButton male;
private JRadioButton female;
private ButtonGroup gengp;
private JLabel dob;
private JComboBox date;
private JComboBox month;
private JComboBox year;
private JLabel add;
private JTextArea tadd;
private JCheckBox term;
private JButton sub;
private JButton reset;
private JTextArea tout;
private JLabel res;
private JTextArea resadd;
private String dates[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12", "13", "14", "15", "16",
"17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30", "31" };
private String months[] = { "Jan", "feb", "Mar", "Apr", "May", "Jun",
"July", "Aug", "Sup", "Oct", "Nov", "Dec" };
private String years[] = { "1995", "1996", "1997", "1998", "1999",
"2000", "2001", "2002", "2003", "2004", "2005",
"2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014",
"2015", "2016", "2017", "2018",
"2019" };
// constructor, to initialize the components
// with default values.
public MyFrame() {
setTitle("Registration Form");
setBounds(300, 90, 900, 600);
```

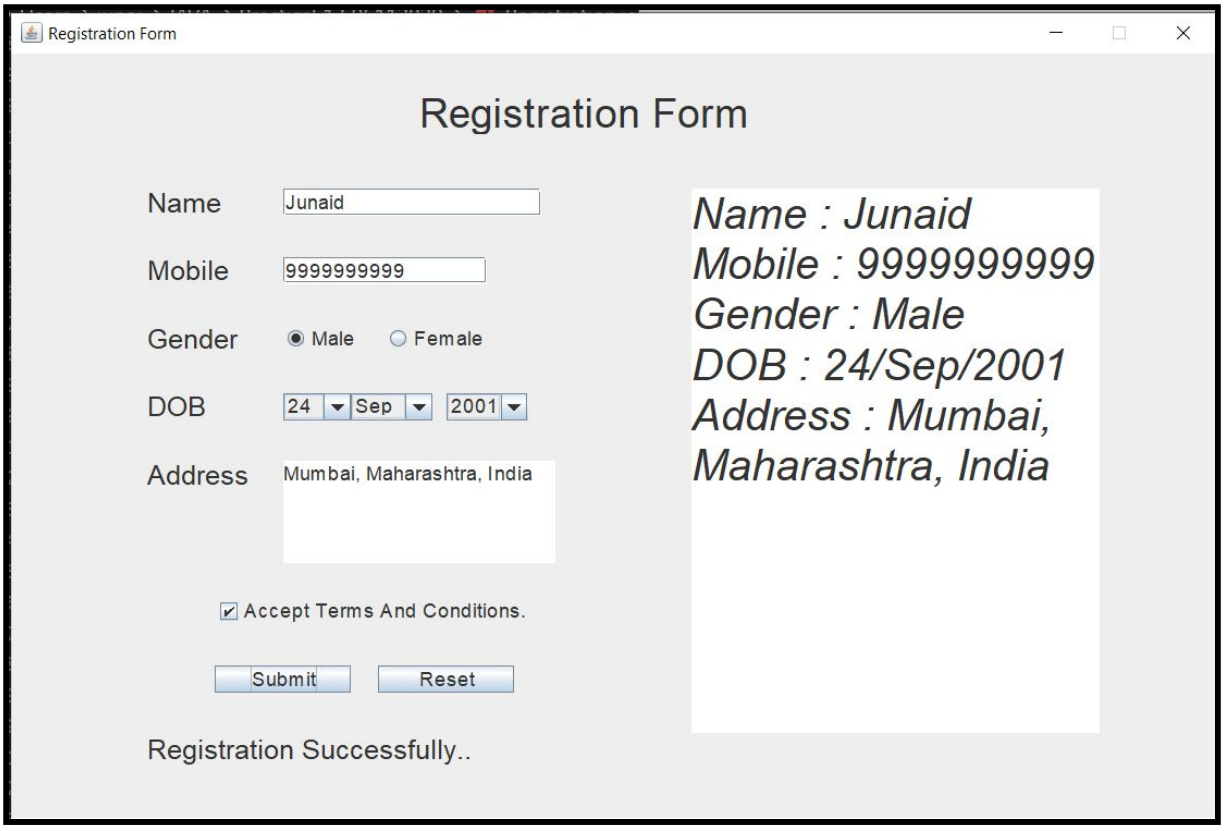
```
setDefaultCloseOperation(EXIT_ON_CLOSE);
setResizable(false);
c = getContentPane();
c.setLayout(null);
title = new JLabel("Registration Form");
title.setFont(new Font("Arial", Font.PLAIN, 30));
title.setSize(300, 30);
title.setLocation(300, 30);
c.add(title);
name = new JLabel("Name");
name.setFont(new Font("Arial", Font.PLAIN, 20));
name.setSize(100, 20);
name.setLocation(100, 100);
c.add(name);
tname = new JTextField();
tname.setFont(new Font("Arial", Font.PLAIN, 15));
tname.setSize(190, 20);
tname.setLocation(200, 100);
c.add(tname);
mno = new JLabel("Mobile");
mno.setFont(new Font("Arial", Font.PLAIN, 20));
mno.setSize(100, 20);
mno.setLocation(100, 150);
c.add(mno);
tmno = new JTextField();
tmno.setFont(new Font("Arial", Font.PLAIN, 15));
tmno.setSize(150, 20);
tmno.setLocation(200, 150);
c.add(tmno);
gender = new JLabel("Gender");
gender.setFont(new Font("Arial", Font.PLAIN, 20));
gender.setSize(100, 20);
gender.setLocation(100, 200);
c.add(gender);
male = new JRadioButton("Male");
male.setFont(new Font("Arial", Font.PLAIN, 15));
male.setSelected(true);
male.setSize(75, 20);
male.setLocation(200, 200);
c.add(male);
female = new JRadioButton("Female");
female.setFont(new Font("Arial", Font.PLAIN, 15));
female.setSelected(false);
female.setSize(80, 20);
female.setLocation(275, 200);
c.add(female);
```

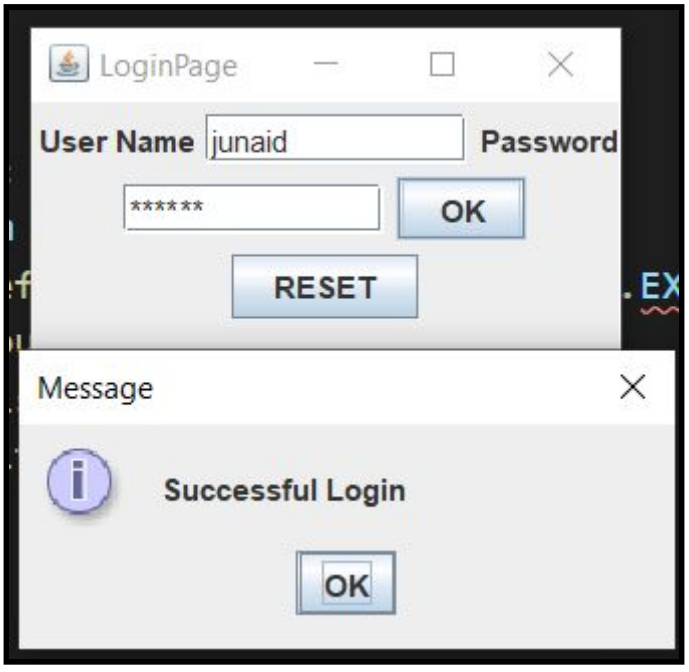
```
gengp = new ButtonGroup();
gengp.add(male);
gengp.add(female);
dob = new JLabel("DOB");
dob.setFont(new Font("Arial", Font.PLAIN, 20));
dob.setSize(100, 20);
dob.setLocation(100, 250);
c.add(dob);
date = new JComboBox(dates);
date.setFont(new Font("Arial", Font.PLAIN, 15));
date.setSize(50, 20);
date.setLocation(200, 250);
c.add(date);
month = new JComboBox(months);
month.setFont(new Font("Arial", Font.PLAIN, 15));
month.setSize(60, 20);
month.setLocation(250, 250);
c.add(month);
year = new JComboBox(years);
year.setFont(new Font("Arial", Font.PLAIN, 15));
year.setSize(60, 20);
year.setLocation(320, 250);
c.add(year);
add = new JLabel("Address");
add.setFont(new Font("Arial", Font.PLAIN, 20));
add.setSize(100, 20);
add.setLocation(100, 300);
c.add(add);
tadd = new JTextArea();
tadd.setFont(new Font("Arial", Font.PLAIN, 15));
tadd.setSize(200, 75);
tadd.setLocation(200, 300);
tadd.setLineWrap(true);
c.add(tadd);
term = new JCheckBox("Accept Terms And Conditions.");
term.setFont(new Font("Arial", Font.PLAIN, 15));
term.setSize(250, 20);
term.setLocation(150, 400);
c.add(term);
sub = new JButton("Submit");
sub.setFont(new Font("Arial", Font.PLAIN, 15));
sub.setSize(100, 20);
sub.setLocation(150, 450);
sub.addActionListener(this);
c.add(sub);
reset = new JButton("Reset");
```

```
reset.setFont(new Font("Arial", Font.PLAIN, 15));
reset.setSize(100, 20);
reset.setLocation(270, 450);
reset.addActionListener(this);
c.add(reset);
// Setting up the text
tout = new JTextArea();
tout.setFont(new Font("Arial", Font.ITALIC, 32));
tout.setSize(300, 400);
tout.setLocation(500, 100);
tout.setLineWrap(true);
tout.setEditable(false);
c.add(tout);
res = new JLabel("");
res.setFont(new Font("Arial", Font.PLAIN, 20));
res.setSize(500, 25);
res.setLocation(100, 500);
c.add(res);
resadd = new JTextArea();
resadd.setFont(new Font("Arial", Font.PLAIN, 15));
resadd.setSize(200, 75);
resadd.setLocation(580, 175);
resadd.setLineWrap(true);
c.add(resadd);
setVisible(true);
}
// method actionPerformed()
// to get the action performed
// by the user and act accordingly
public void actionPerformed(ActionEvent e) {
if (e.getSource() == sub) {
if (term.isSelected()) {
String data1;
String data = "Name : " + tname.getText() + "\n" + "Mobile : " +
tmno.getText() + "\n";
if (male.isSelected())
data1 = "Gender : Male" + "\n";
else
data1 = "Gender : Female" + "\n";
String data2 = "DOB : " + (String) date.getSelectedItem() + "/" +
(String) month.getSelectedItem() + "/"
+ (String) year.getSelectedItem() + "\n";
String data3 = "Address : " + tadd.getText();
tout.setText(data + data1 + data2 + data3);
tout.setEditable(false);
res.setText("Registration Successfully..");
```

```
} else {
tout.setText("");
resadd.setText("");
res.setText("Please accept the" + " terms & conditions..");
}
}
else if (e.getSource() == reset) {
String def = "";
tname.setText(def);
tadd.setText(def);
tmno.setText(def);
res.setText(def);
tout.setText(def);
term.setSelected(false);
date.setSelectedIndex(0);
month.setSelectedIndex(0);
year.setSelectedIndex(0);
resadd.setText(def);
}
}
}
// Driver Code
class Registration {
public static void main(String[] args) throws Exception {
MyFrame f = new MyFrame();
}
}
```

OUTPUT :


Registration Form
Registration Form
Name
Junaid
Mobile
9999999999
Gender
Male
Female
DOB
24
Sep
2001
Address
Mumbai, Maharashtra, India
Accept Terms And Conditions.
Submit
Reset
Registration Successfully..
Name : Junaid
Mobile : 9999999999
Gender : Male
DOB : 24/Sep/2001
Address : Mumbai, Maharashtra, India


LoginPage
User Name
junaid
Password
******
OK
RESET
Message
Successful Login
OK