NAME: JUNAID GIRKAR  
SAP ID: 60004190057

DIV: SE COMPS A-3

**JAVA**

**PRACTICALS**

**SEMISTER 3**

EXPERIMENT 1

**AIM:** To implement Java Program Structures & Simple Programs

Q1. WAP to display hello Message on screen.

CODE:

import java.util.\*;

class hello {

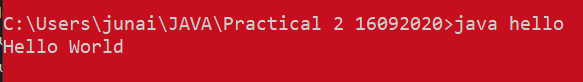
public static void main(String args[]) {

System.out.println("Hello World");

}

}

OUTPUT:



Q2. Write a Java program that reads a positive integer from 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++;

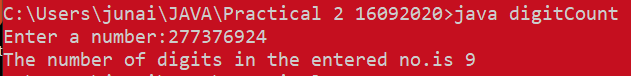
}

System.out.print("The number of digits in the entered no.is "+count);

}

}

OUTPUT:



EXPERIMENT 2

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);*

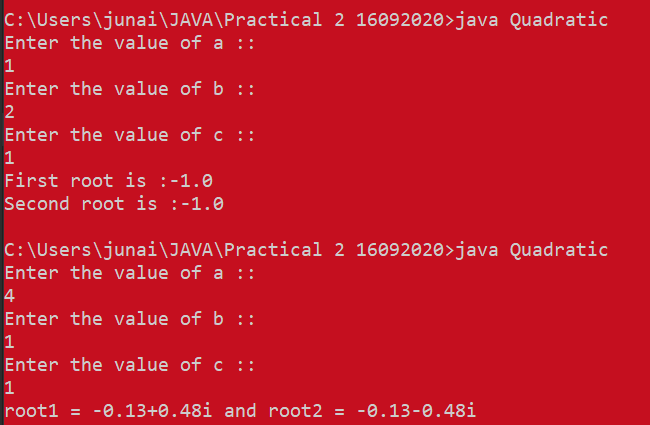
*System.out.format("root1 = %.2f+%.2fi and root2 = %.2f-%.2fi", realPart, imaginaryPart, realPart, imaginaryPart);*

}

}

}

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;*

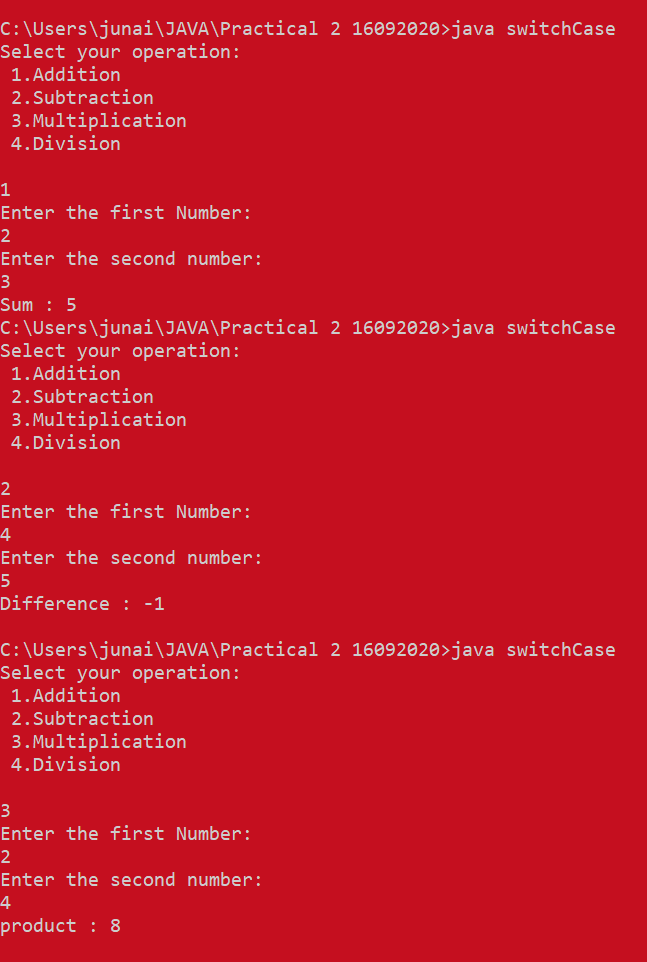
*System.out.println("Division : " + division);*

*break;*

*default:*

*System.out.println("Invalid Choice");*

*} } }*

*OUTPUT:*

***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;

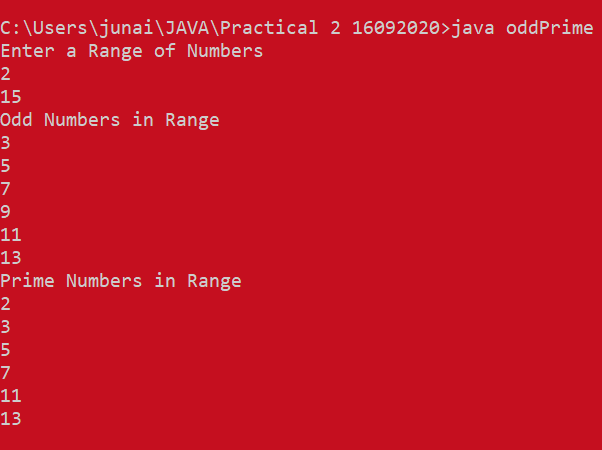
}

ob.close();

}

*}*

*OUTPUT:*

**

***Q5) WAP to display the following patterns:***

***CODE:***

*class Patern {*

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));

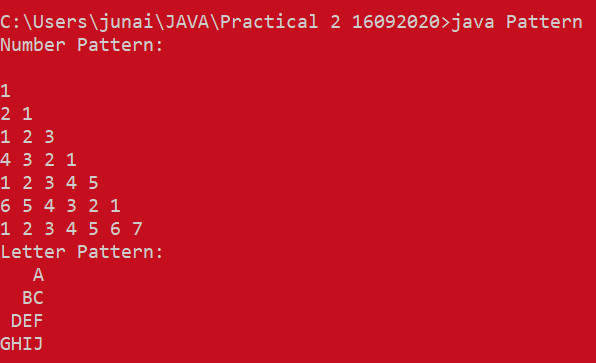
a++;

} 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 {*

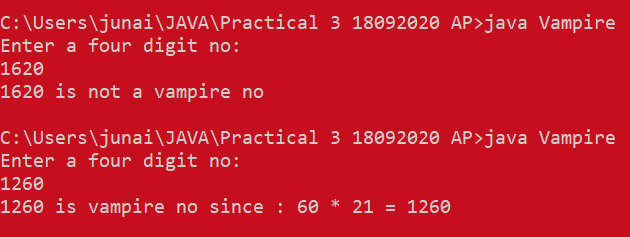
*System.out.println(no + " is not a vampire no");*

*}*

*}*

*}*

*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]);

}

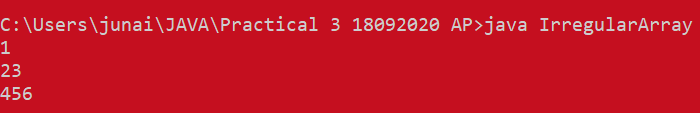
System.out.println();

}

}

}

OUTPUT:

**

Q3.



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]);

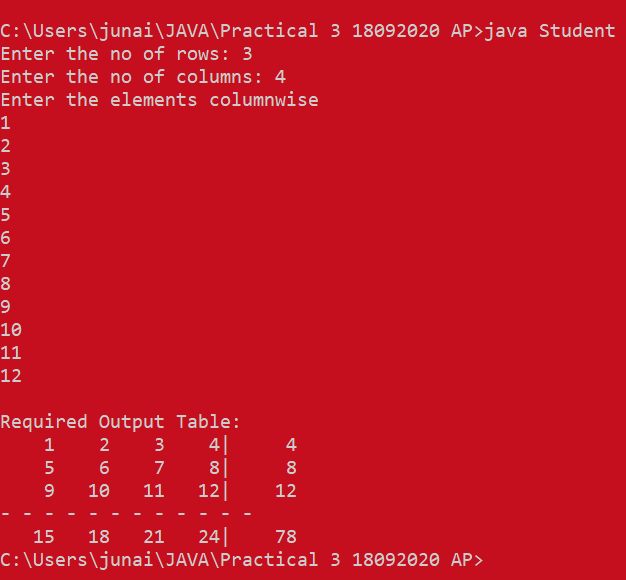
}

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: ");*

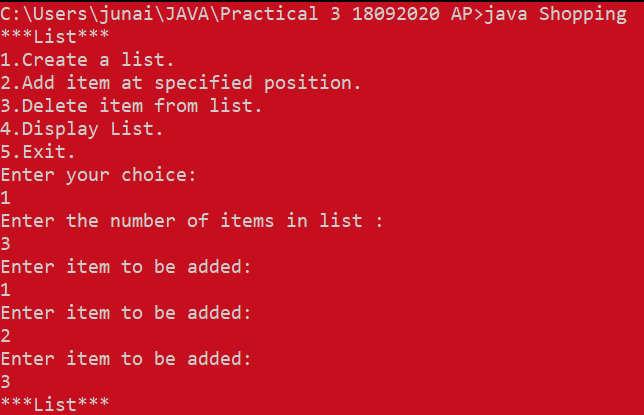
*ch = in.nextInt();*

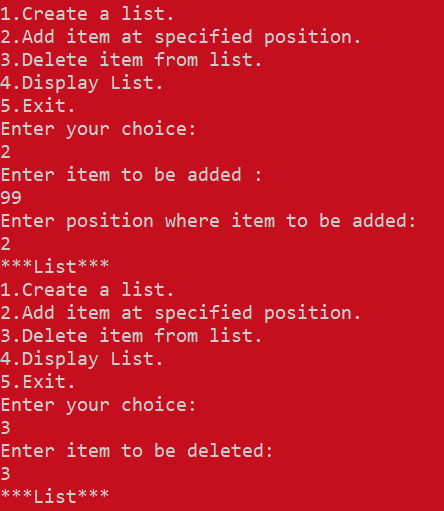
*}*

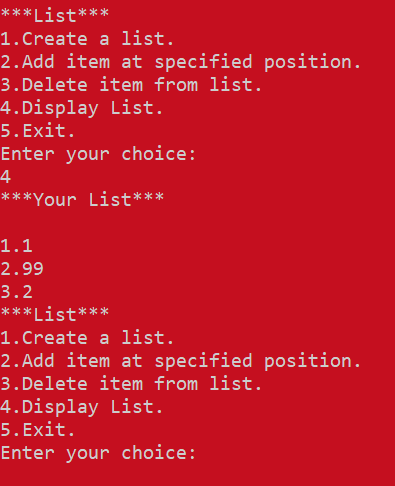
*in.close();*

*}*

*}*

*OUTPUT:*





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

CODE:  
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++) {

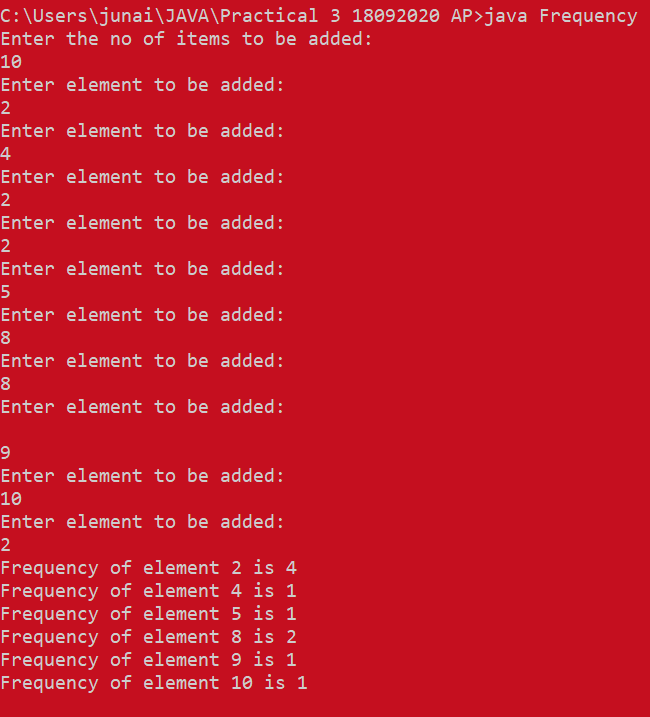
if (fr[i] != visited)

System.out.println("Frequency of element " + v.elementAt(i) + " is " + fr[i]);

}

}

}

OUTPUT:

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 current character doesn't match

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;

}

}

// Check if previous unmatched of string1

// is equal to curr unmatched of string2

// and also check for curr unmatched character,

// if both are same, then return true

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

}

// Driver method

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();

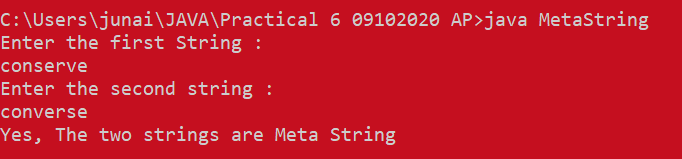
System.out.println(areMetaStrings(string1, string2) ? "Yes, The two strings are Meta String"

: "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));

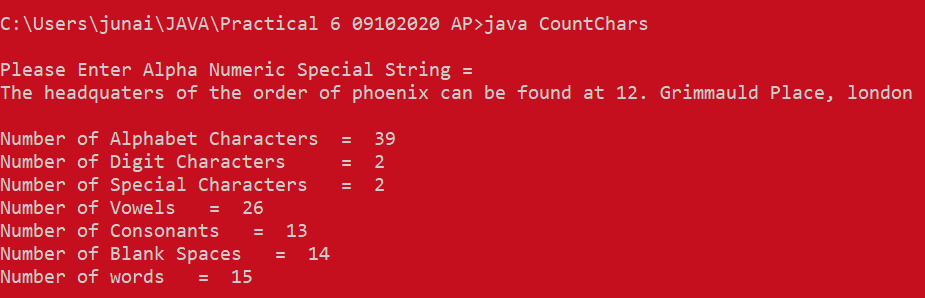
System.out.println("Number of Blank Spaces = " + blankSpace);

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();

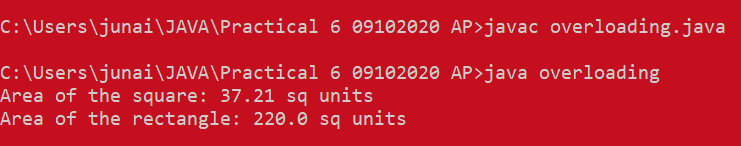
obj.calculateArea(6.1f);

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 XY

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');

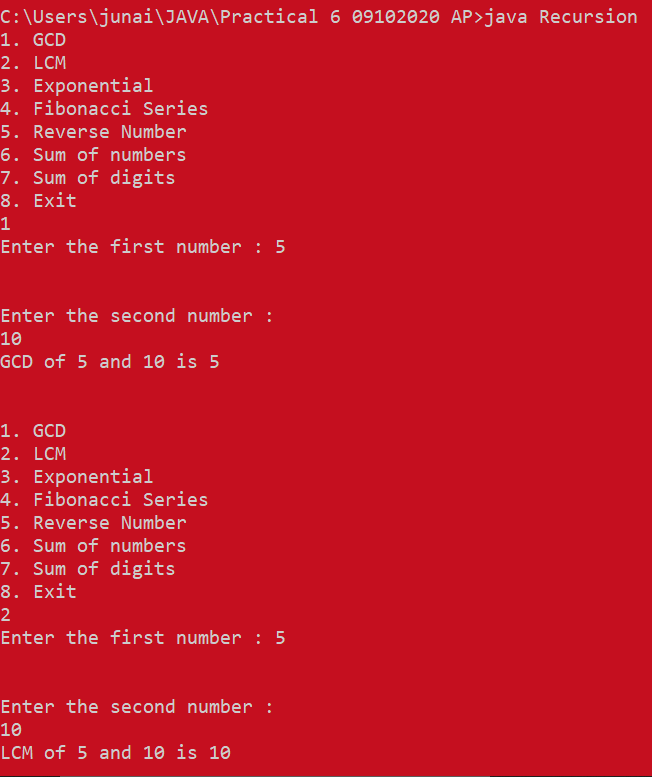
}

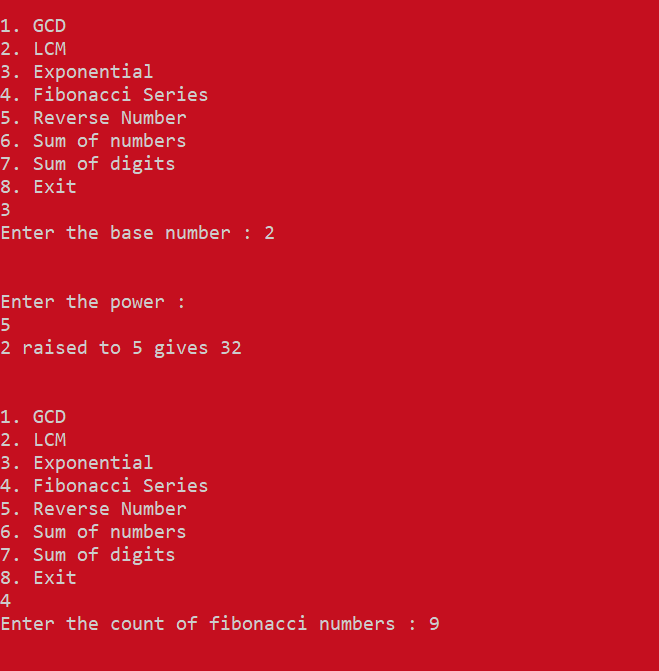
}

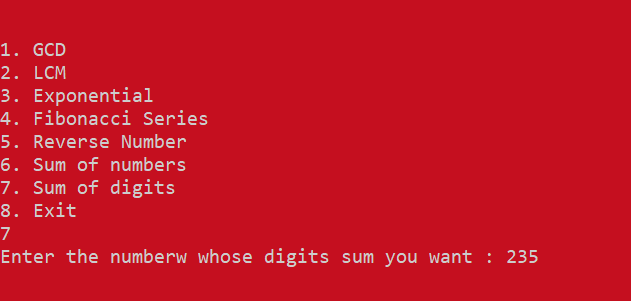
} while (choice != 8);

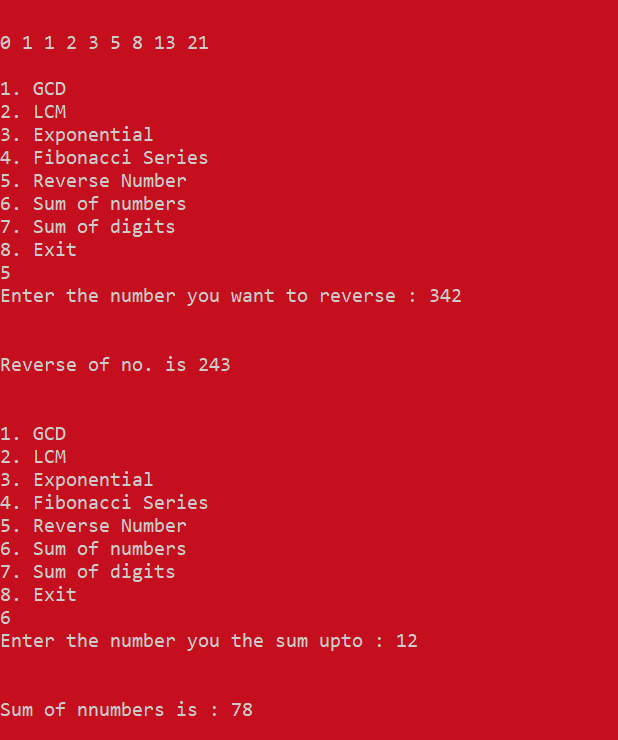
}

}

OUTPUT:







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 students 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 Succussfully 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;

}

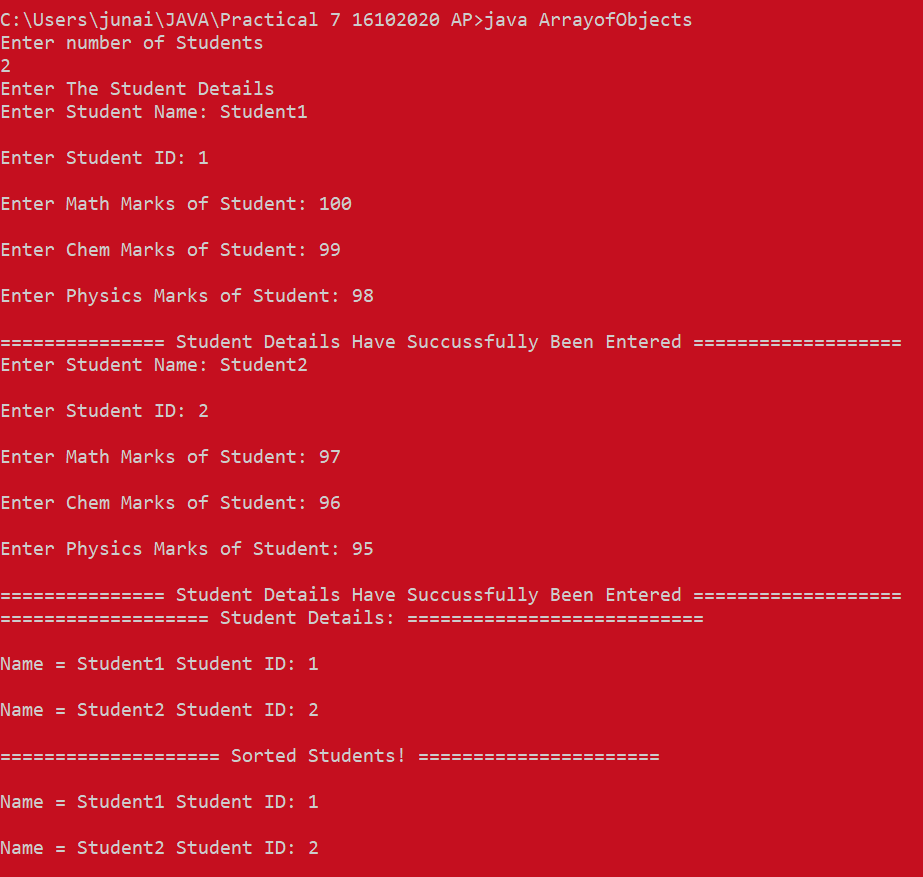
public long getChemMarks() {

return this.chem\_marks;

}

}

OUTPUT:



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);

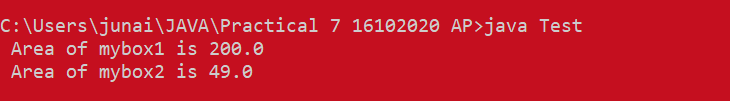
// get area of second box

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 voliume. If z refers to an object of any other class, then size(z) returns -1. Use 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));

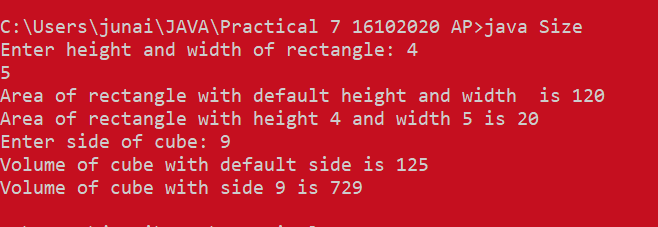
Cube cb = new Cube(x);

System.out.println("Volume of cube with side " + x + " is " + size(cb));

}

}

OUTPUT:



EXPERIMENT 9

AIM: **To implement Abstract classes**

**Q1.** Write a ***abstract class*** program to calculate area of 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);

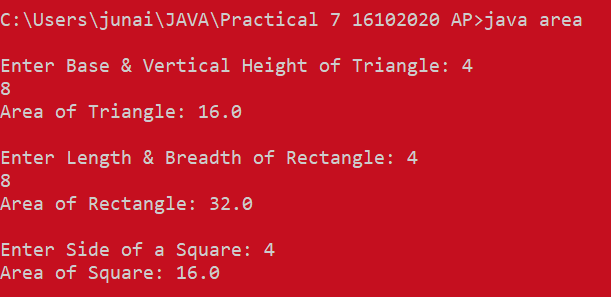
System.out.print("\nEnter Side of a Square: ");

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 member as rollno, Test class has members as sem1\_marks and sem2\_marks and Result class has member 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 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();**

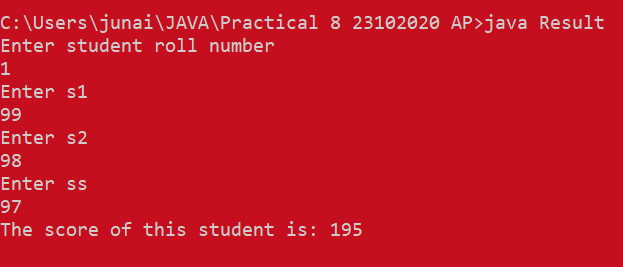
**}**

**}**

**interface Sports {**

**void score();**

**}**

OUTPUT:

EXPERIMENT 11

AIM: **To implement Package**

**CODE:**

PACKAGE (..\letmeadd\Calculator.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));

}

}

import letmeadd.Calculator;

public class Demo {

    public static void main(String args[]) {

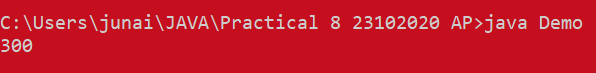
        Calculator obj = new Calculator();

        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 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++;

            }

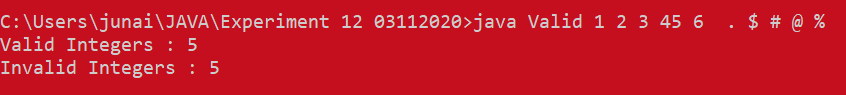
        }

        System.out.println("Valid Integers : " + valid + "\nInvalid Integers : " + invalid);

    }

}

**OUTPUT:**

****

**Q2.** Write a Java Program to calculate the Result. Result should consist of name, seatno, 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);**

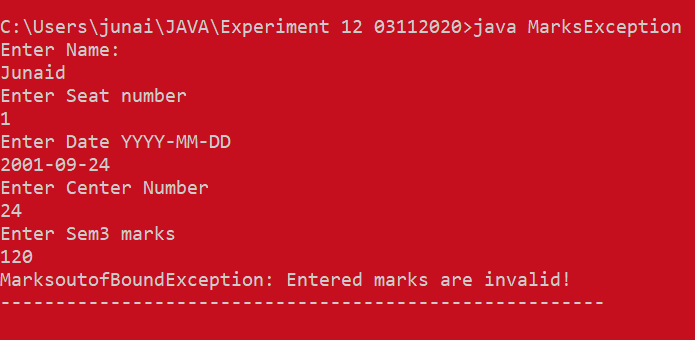
**}**

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

**}**

**}**

**}**

**OUTPUT:**