**ASSIGNMENT 1**

**1. Write a Java program to count letters, spaces, numbers and other characters**

**n an input string.**

**Code:**

import java.io.\*;

import java.util. \*;

class count

{

public static void main(String args[])throws IOException

{

String str;

char ch;

int space=0,letter=0,other=0,number=0;

System.out.println("Enter the String:");

Scanner sc=new Scanner(System.in);

str=sc.nextLine();

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

{

ch=str.charAt(i);

if(Character.isDigit(ch))

number++;

else if(Character.isLetter(ch))

letter++;

else if(Character.isWhitespace(ch))

space++;

else

other++;

}

System.out.println("Letter:"+letter);

System.out.println("Space:"+space);

System.out.println("Number:"+number);

System.out.println("Other:"+other);

}

}

**Output:**

Enter the String:

Aa kiu, I swd skieo 236587. GH kiu: sieo??

25.33Letter:23

Space:8

Number:6

Other:5

**2. Write a menu driven java program to find**

**i. If the string length is less than 3 use "#" as substitute characters.**

**ii. Concatenate two strings**

**iii. Extract Substring**

**iv. Reverse a string**

**Code:**

import java.io.\*;

import java.util.\*;

class Stringfunction {

public static void main(String[] args)throws IOException {

String str1, str2;

int choice;

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("String Manipulation Program");

System.out.println("1. Substitute characters");

System.out.println("2. Concatenate strings");

System.out.println("3. Extract substring");

System.out.println("4. Reverse a string");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // to take newline character

switch (choice) {

case 1:

System.out.println("Enter the String :");

str1 = scanner.nextLine();

if (str1.length() < 3)

str1 = "#";

System.out.println("Substituted String: " + str1);

break;

case 2:

System.out.print("Enter the first string: ");

str1 = scanner.nextLine();

System.out.print("Enter the second string: ");

str2 = scanner.nextLine();

String str3 = str1.concat(str2);

System.out.println("Concatenated string: " + str3);

break;

case 3:

System.out.print("Enter a string: ");

str1 = scanner.nextLine();

System.out.print("Enter the starting index: ");

int startIndex = scanner.nextInt();

System.out.print("Enter the ending index: ");

int endIndex = scanner.nextInt();

String substring = str1.substring(startIndex, endIndex);

System.out.println("Extracted substring: " + substring);

break;

case 4:

System.out.print("Enter a string: ");

str1 = scanner.nextLine();

String reversedString = new StringBuilder(str1).reverse().toString();

System.out.println("Reversed string: " + reversedString);

break;

case 5:

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

}

}

}

**Output:**

String Manipulation Program

1. Substitute characters

2. Concatenate strings

3. Extract substring

4. Reverse a string

5. Exit

Enter your choice: 1

Enter the String :

vv

Substituted String: #

String Manipulation Program

1. Substitute characters

2. Concatenate strings

3. Extract substring

4. Reverse a string

5. Exit

Enter your choice: 2

Enter the first string: enterprise

Enter the second string: java

Concatenated string: enterprisejava

String Manipulation Program

1. Substitute characters

2. Concatenate strings

3. Extract substring

4. Reverse a string

5. Exit

Enter your choice: 3

Enter a string: enterprise

Enter the starting index: 2

Enter the ending index: 5

Extracted substring: ter

String Manipulation Program

1. Substitute characters

2. Concatenate strings3. Extract substring

4. Reverse a string

5. Exit

Enter your choice: 4

Enter a string: java

Reversed string: avaj

**3. Write a Java program to swap the first and last elements of an array and create another array.**

Code:

import java.io.\*;

import java.util.\*;

class ArraySwap {

public static void main(String args[]) throws IOException {

int n, temp;

System.out.println("Enter the Number of elements:");

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

int[] a1 = new int[n];

int[] a2 = new int[n];

System.out.println("Enter "+n+" elements:");

for (int i = 0; i < n; i++)

a1[i] = sc.nextInt();

temp = a1[0]; //swapping

a1[0] = a1[a1.length - 1];

a1[a1.length - 1] = temp;

for (int i = 0; i < n; i++) {

a2[i] = a1[i]; //copying to another array

}

System.out.println("Elements After Swapping:");

for (int i = 0; i < n; i++) {

System.out.print(a2[i]+" ");

}

}

}

**Output:**

Enter the Number of elements:

5

Enter 5 elements:

8 6 9 7 4

Elements After Swapping:

4 6 9 7 8

**4. Write a Java program that then reads an integer and calculates the sum of its digits and writes the number of each digit of the sum in English.**

Code:

import java.io.\*;

import java.util.\*;

class DigitSum {

public static void main(String args[]) throws IOException {

int n, temp,sum=0;

System.out.println("Enter the Number:");

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

temp=n;

while(temp!=0)

{

int digit=temp%10;

sum+=digit;

temp=temp/10;

}

String[] names={"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};

String sum\_digit="";

while(sum!=0)

{

int digit=sum%10;

sum\_digit=names[digit]+" "+sum\_digit;

sum=sum/10;

}

System.out.println("Sum of digits in English: " +sum\_digit);

}

}

**Output:**

Enter the Number:

99

Sum of digits in English: one eight

Enter the Number:

82

Sum of digits in English: one zero

**5. Java code to calculate the CGPA (Cumulative Grade Point Average ). Note: Students marks in 5 subjects can be considered for calculating CGPA.**

Code:

import java.io.\*;

import java.util.\*;

class CalculateCgpa {

public static void main(String args[]) throws IOException {

int n, total=0;

double cgpa;

int[] student = new int[6];

Scanner sc = new Scanner(System.in);

for (int i = 1; i <= 5; i++)

{

System.out.println("Enter Subject "+i+" Marks: ");

student[i] = sc.nextInt();

}

for (int i = 1; i <= 5; i++)

total+=student[i];

cgpa=(total/(5.0 \* 100)) \* 10;

System.out.print("CGPA: "+cgpa);

}

}

**Output:**

Enter Subject 1 Marks: 90

Enter Subject 2 Marks: 95

Enter Subject 3 Marks: 85

Enter Subject 4 Marks: 92

Enter Subject 5 Marks: 82

CGPA: 8.88

**6. Write a java program to find the transpose of a matrix.**

**Code:**

import java.io.\*;

import java.util.\*;

class TransposeMatrix {

public static void main(String args[]) throws IOException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Number of Rows in the Matrix:");

int rows = sc.nextInt();

System.out.println("Enter Number of Columns in the Matrix:");

int columns = sc.nextInt();

int[][] matrix = new int[rows][columns];

System.out.println("Enter the Elements of the Matrix:");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

matrix[i][j] = sc.nextInt();

}

}

int[][] transpose = new int[columns][rows];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

transpose[j][i] = matrix[i][j];

}

}

System.out.println("Transpose of the Matrix:");

for (int i = 0; i < columns; i++) {

for (int j = 0; j < rows; j++) {

System.out.print(transpose[i][j] + " ");

}

System.out.println();

}

}

}

**Output:**

Enter Number of Rows in the Matrix:

3

Enter Number of Columns in the Matrix:

3

Enter the Elements of the Matrix:

1 2 3

4 5 6

7 8 9

Transpose of the Matrix:

1 4 7

2 5 8

3 6 9

**7. Write a java program to find the second largest number in an array.**

**Code:**

import java.io.\*;

import java.util.\*;

class SecondLargestNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n, i;

System.out.println("Enter the number of elements in an array:");

n = sc.nextInt();

int[] a = new int[n];

System.out.println("Enter array elements:");

for (i = 0; i < n; i++) {

a[i] = sc.nextInt();

}

int largest = Integer.MIN\_VALUE;

int secondLargest = Integer.MIN\_VALUE;

for (i = 0; i < n; i++) {

if (a[i] > largest) {

secondLargest = largest;

largest = a[i];

} else if (a[i] > secondLargest && a[i] < largest) {

secondLargest = a[i];

}

}

if(n<2)

System.out.println("Please Enter at least two number");

else

System.out.println("The second largest number in the array is: " + secondLargest);

}

}

**Output:**

Enter the number of elements in an array:

6

Enter array elements:

5 8 9 3 2 5

The second largest number in the array is: 8

**8. Write a java program to find the index of the largest number in an array**

**Code:**

import java.io.\*;

import java.util.\*;

class LargestIndex {

public static void main(String args[]) throws IOException {

int n,index=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter Number of elements in the array:");

n=sc.nextInt();

int[] a = new int[n];

System.out.println("Enter "+n+" Array elements: ");

for (int i = 0; i<n; i++)

{

a[i] = sc.nextInt();

}

for (int i = 0; i<n; i++) {

if(a[i]>a[index])

index=i;

}

System.out.print("Largest Number Index: "+index);

}

}

**Output:**

Enter Number of elements in the array:

6

Enter 6 Array elements:

5 6 8 2 9 7

Largest Number Index: 4

**9.Write a java program to find electricity bill. Note: if you consume between 300 and 500 units, then charges are fixed as 7.75 Rupees for Units.**

**Code:**

import java.io.\*;

import java.util.\*;

class EletricityBill {

public static void main(String args[]) throws IOException {

int units;

double bill;

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of units consumed: ");

units=sc.nextInt();

if(units>=300 && units<=500)

bill=units\*7.75;

else {

System.out.println("Please Enter the Units Correctly");

return;

}

System.out.println("Electricity Bill Amount: " + bill);

}

}

**Output:**

Enter the number of units consumed: 400

Electricity Bill Amount: 3100.0

Enter the number of units consumed: 525

Please Enter the Units Correctly

**10. Write a java program to print below pattern using stars.**

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**Code:**

import java.io.\*;

import java.util.\*;

class Pattern {

public static void main(String[] args) {

for(int i=5;i>=1;i--)

{

for(int j=1;j<=i;j++)

{

System.out.print("\*");

}

System.out.println();

}

}

}

**Output:**

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**11. Write a java program to find Nth Fibonacci number.**

**Code:**

import java.io.\*;

import java.util.\*;

class Fibonacci {

public static void main(String args[]) throws IOException {

int n;

System.out.println("Enter the Number:");

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

if (n < 0)

System.out.println("Please Enter a Proper Number");

else {

int f1 = 0, f2 = 1, f3;

for (int i = 0; i < n - 1; i++) {

f3 = f1 + f2;

f1 = f2;

f2 = f3;

}

System.out.println("Nth Fibonacci Number is: "+f1);

}

}

}

**Output:**

Enter the Number:10

Nth Fibonacci Number is: 34

**12. Write a java program to print N Fibonacci numbers.**

import java.io.\*;

import java.util. \*;

class fib

{

public static void main(String args[])throws IOException

{

int n;

System.out.println("Enter the Number:");

Scanner sc=new Scanner(System.in);

n=sc.nextInt();

if(n<0)

System.out.println("Please Enter Proper number");

else

{

int f1=0,f2=1,f3;

System.out.println("Fibonaci Numbers are:");

for(int i=0;i<n;i++)

{

System.out.print(f1+" ");

f3=f1+f2;

f1=f2;

f2=f3;

}

}

}

}

**Output:**

Enter the Number:10

Fibonaci Numbers are:

0 1 1 2 3 5 8 13 21 34

**13. Write a java program to Reverse sentence.**

import java.io.\*;

import java.util.\*;

class ReverseString {

public static void main(String args[]) throws IOException {

String str, rev = "";

System.out.println("Enter the String:");

Scanner sc = new Scanner(System.in);

str = sc.nextLine();

for (int i = str.length() - 1; i >= 0; i--) {

rev = rev + str.charAt(i);

}

System.out.println("Reversed Sentnece is: "+rev);

}

}

**Output:**

Enter the String:

Enterprise java

Reversed Sentnece is: avaj esirpretnE

**14. Write a java program to generate random number**

**Code:**

import java.io.\*;

import java.util.\*;

class RandomNumbers {

public static void main(String[] args) {

int n;

Scanner sc = new Scanner(System.in);

System.out.println("Enter Number of random Numbers :");

n=sc.nextInt();

Random rand=new Random();

System.out.print(n+" random Numbers are:");

for(int i=0;i<n;i++)

{

System.out.println(rand.nextInt(100)); //generating random numbers upto 99

}

}

}

**Output:**

Enter Number of random Numbers :

5

5 random Numbers are:

21

25

46

83

32

**15. Write a java program to check for leap year**

**Code:**

import java.io.\*;

import java.util.\*;

class LeapYear {

public static void main(String[] args) {

int year;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Year:");

year = sc.nextInt();

if((year%4==0 && year%100!=0) ||year%400==0)

System.out.println(year + " is a leap year.");

else

System.out.println(year + " is not a leap year.");

}

}

**Output:**

Enter the Year:

2020

2020 is a leap year.

Enter the Year:

2023

2023 is not a leap year.

**16. Write a java program to print Characters in a String and their ASCII Values**

**Code:**

import java.io.\*;

import java.util.\*;

class Asciivalue {

public static void main(String[] args) {

String str;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the String:");

str = sc.nextLine();

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

char c=str.charAt(i);

int ascii=(int)c;

System.out.println("Character: " + c + " ASCII Value: " + ascii);

}

}

}

**Output**:

Enter the String:

Java

Character: J ASCII Value: 74

Character: a ASCII Value: 97

Character: v ASCII Value: 118

Character: a ASCII Value: 97

**17. Write a java program to check Perfect Number**

**Code:**

import java.io.\*;

import java.util.\*;

class PerfectNumber {

public static void main(String[] args) {

int n,sum=0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Number:");

n = sc.nextInt();

for (int i = 1; i <= n/2; i++) {

if (n%i==0) {

sum+=i;

}

}

if(n==sum)

System.out.println(n+" is a perfect number ");

else

System.out.println(n+" is not a perfect number ");

}

}

**Output:**

Enter the Number:

28

28 is a perfect number

**Note: Perfect number is equal to sum of its proper divisor**

**Eg: number 28. its proper divisor are (1,2,4,7,14) equal to 28**

**18. Write a java program to Binary search in an array.**

Code:

import java.io.\*;

import java.util.\*;

class BinarySearch {

public static void main(String[] args) {

int n, i, j, x, temp;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of elements in an array:");

n = sc.nextInt();

int[] a = new int[n];

System.out.println("Enter array elements:");

for (i = 0; i < n; i++) {

a[i] = sc.nextInt();

}

for (i = 0; i < n - 1; i++) {

for (j = 0; j < n - i - 1; j++) {

if (a[j] > a[j + 1]) {

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

System.out.println("Array elements after Sorting:");

for (i = 0; i < n; i++)

System.out.print(a[i]+" ");

int mid, l, r;

l = 0;

r = n - 1;

System.out.println("Enter the element to be searched : ");

x = sc.nextInt();

while (l <= r) {

mid = (l + r) / 2;

if (x == a[mid]) {

System.out.println("Element found at index " + mid);

return;

}

if (x < a[mid])

r = mid - 1; else

l = mid + 1;

}

System.out.println("Element not found");

}

}

**Output:**

Enter the number of elements in an array:

5

Enter array elements:

5 6 9 3 2

Array elements after Sorting:

2 3 5 6 9

Enter the element to be searched :

3

Element found at index 1

**19. Write a java program to find gcd.**

**Code:**

import java.io.\*;

import java.util.\*;

class FindGCD {

public static void main(String[] args) {

int num1,num2;

Scanner sc = new Scanner(System.in);

System.out.println("Enter First Number :");

num1=sc.nextInt();

System.out.println("Enter Second Number :");

num2=sc.nextInt();

int gcd = findGCD(num1, num2);

System.out.println("The GCD of " + num1 + " and " + num2 + " is: " + gcd);

}

public static int findGCD(int num1, int num2) {

while (num2 != 0) {

int temp = num2;

num2 = num1 % num2;

num1 = temp;

}

return num1;

}

}

**Output:**

Enter First Number :

20

Enter Second Number :

10

The GCD of 20 and 10 is: 10

**20. Write a java program to find Vowels and consonants in a string**

**Code:**

import java.io.\*;

import java.util.\*;

class Vowels {

public static void main(String[] args) {

String str;

int vowels = 0, consonants = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the String:");

str = sc.nextLine();

str = str.toLowerCase();

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

char ch = str.charAt(i);

if (ch >= 'a' && ch <= 'z') {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowels++;

} else {

consonants++;

}

}

}

System.out.println("Number of vowels: " + vowels);

System.out.println("Number of consonants: " + consonants);

}

}

**Output:**

Enter the String:

Enterprise java

Number of vowels: 6

Number of consonants: 8

**ASSIGNMENT 2**

12-02-2024

Apply the following to any of the questions of Assignment 1

**1. Define atleast two classes**

**Code:**

import java.io.\*;

import java.util. \*;

class fib

{

public static void fibonacci(int n)

{

if(n<0)

System.out.println("Please Enter Proper number");

else

{

int f1=0,f2=1,f3;

for(int i=0;i<n;i++)

{

System.out.println(f1);

f3=f1+f2;

f1=f2;

f2=f3;

}

}

}

}

class main

{

public static void main(String args[])throws IOException

{

int n;

System.out.println("Enter the Number:");

Scanner sc=new Scanner(System.in);

n=sc.nextInt();

fib.fibonacci(n);

}

}

**Output:**

Enter the Number:5

Fibonacci Numbers are:

0

1

1

2

3

**2. To read the input from the user, define either methods or**

**Constructors**

**Code:**

import java.io.\*;

import java.util.\*;

class LeapYear {

int year;

LeapYear()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Year:");

year = sc.nextInt();

find(year);

}

public static void find(int year)

{

if((year%4==0 && year%100!=0) ||year%400==0)

System.out.println(year + " is a leap year.");

else

System.out.println(year + " is not a leap year.");

}

}

class main

{

public static void main(String[] args) {

LeapYear l=new LeapYear();

}

}

**Output:**

Enter the Year:

2025

2025 is not a leap year.

**3. Overload the methods and constructors in certain situations**

**Code:**

import java.io.\*;

import java.util.\*;

class Asciivalue {

String str;

Asciivalue() {

}

Asciivalue(String s) {

str = s;

}

void compute( String str)

{

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

char c = str.charAt(i);

int ascii = (int) c;

System.out.println("Character: " + c + " ASCII Value: " + ascii);

}

}

void compute()

{

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

char c = str.charAt(i);

int ascii = (int) c;

System.out.println("Character: " + c + " ASCII Value: " + ascii);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the String:");

String str = sc.nextLine();

Asciivalue a = new Asciivalue(str);

a.compute(str);

}

}

**Output:**

Enter the String:

JavaEnterprise

Character: J ASCII Value: 74

Character: a ASCII Value: 97

Character: v ASCII Value: 118

Character: a ASCII Value: 97

Character: E ASCII Value: 69

Character: n ASCII Value: 110

Character: t ASCII Value: 116

Character: e ASCII Value: 101

Character: r ASCII Value: 114

Character: p ASCII Value: 112

Character: r ASCII Value: 114

Character: i ASCII Value: 105

Character: s ASCII Value: 115

Character: e ASCII Value: 101

**4. Use this keyword to differentiate normal variables and instance**

**Variables**

**Code:**

import java.io.\*;

import java.util.\*;

class RandomNumbers {

int n;

RandomNumbers(int n) {

this.n=n;

}

}

class main {

public static void main(String[] args) {

int n;

Scanner sc = new Scanner(System.in);

System.out.println("Enter Number of random Numbers :");

n=sc.nextInt();

RandomNumbers r = new RandomNumbers(n);

Random rand=new Random();

System.out.print(n+" random Numbers are:");

for(int i=0;i<n;i++)

{

System.out.println(rand.nextInt(100));

}

}

}

**Output:**

Enter Number of random Numbers :

5

5 random Numbers are:

72

6

64

77

26

**5. Pass objects as parameters to methods and make the methods to**

**return the objects of the class (Write a program to perform complex**

**arithmetic)**

**Code:**

import java.io.\*;

import java.util.\*;

class Complex {

int real;

int imaginary;

Complex(int r, int i) {

real = r;

imaginary = i;

}

Complex add(Complex c1, Complex c2) {

return new Complex(c1.real + c2.real, c1.imaginary + c2.imaginary);

}

Complex subtract(Complex c1, Complex c2) {

return new Complex(c1.real - c2.real, c1.imaginary - c2.imaginary);

}

Complex multiplication(Complex c1, Complex c2) {

int realPart = c1.real \* c2.real - c1.imaginary \* c2.imaginary;

int imaginaryPart = c1.real \* c2.imaginary + c1.imaginary \* c2.real;

return new Complex(realPart, imaginaryPart);

}

Complex division(Complex c1, Complex c2) {

int denominator = c2.real \* c2.real + c2.imaginary \* c2.imaginary;

int realPart = (c1.real \* c2.real + c1.imaginary \* c2.imaginary) / denominator;

int imaginaryPart = (c1.imaginary \* c2.real - c1.real \* c2.imaginary) / denominator;

return new Complex(realPart, imaginaryPart);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Real Part:");

int real = sc.nextInt();

System.out.println("Enter the Imaginary Part:");

int imaginary = sc.nextInt();

Complex c1 = new Complex(real, imaginary);

System.out.println("Enter the Real Part:");

real = sc.nextInt();

System.out.println("Enter the Imaginary Part:");

imaginary = sc.nextInt();

Complex c2 = new Complex(real, imaginary);

Complex sum = c1.add(c1, c2);

System.out.println("Sum: " + sum.real + " + " + sum.imaginary + "i");

Complex difference = c1.subtract(c1, c2);

System.out.println("Difference: " + difference.real + " + " + difference.imaginary + "i");

Complex product = c1.multiplication(c1, c2);

System.out.println("Product: " + product.real + " + " + product.imaginary + "i");

Complex quotient = c1.division(c1, c2);

System.out.println("Quotient: " + quotient.real + " + " + quotient.imaginary + "i");

}

}

**Or:**

import java.io.\*;

import java.util.\*;

class Complex {

int real;

int imaginary;

Complex(int r, int i) {

real = r;

imaginary = i;

}

Complex add(Complex c2) {

return new Complex(real + c2.real, imaginary + c2.imaginary);

}

Complex subtract(Complex c2) {

return new Complex(real - c2.real, imaginary - c2.imaginary);

}

Complex multiplication(Complex c2) {

int realPart = real \* c2.real - imaginary \* c2.imaginary;

int imaginaryPart = real \* c2.imaginary + imaginary \* c2.real;

return new Complex(realPart, imaginaryPart);

}

Complex division(Complex c2) {

int denominator = c2.real \* c2.real + c2.imaginary \* c2.imaginary;

int realPart = (real \* c2.real + imaginary \* c2.imaginary) / denominator;

int imaginaryPart = (imaginary \* c2.real - real \* c2.imaginary) / denominator;

return new Complex(realPart, imaginaryPart);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Real Part:");

int real = sc.nextInt();

System.out.println("Enter the Imaginary Part:");

int imaginary = sc.nextInt();

Complex c1 = new Complex(real, imaginary);

System.out.println("Enter the Real Part:");

real = sc.nextInt();

System.out.println("Enter the Imaginary Part:");

imaginary = sc.nextInt();

Complex c2 = new Complex(real, imaginary);

Complex sum = c1.add(c2);

System.out.println("Sum: " + sum.real + " + " + sum.imaginary + "i");

Complex difference = c1.subtract(c2);

System.out.println("Difference: " + difference.real + " + " + difference.imaginary + "i");

Complex product = c1.multiplication(c2);

System.out.println("Product: " + product.real + " + " + product.imaginary + "i");

Complex quotient = c1.division(c2);

System.out.println("Quotient: " + quotient.real + " + " + quotient.imaginary + "i");

}

}

**Output:**

Enter the Real Part:

3

Enter the Imaginary Part:

-4

Enter the Real Part:

4

Enter the Imaginary Part:

5

Sum: 7 + 1i

Difference: -1 + -9i

Product: 32 + -1i

Quotient: 0 + 0i

**6. Define inner classes or nested classes and access the members**

**Code:**

import java.io.\*;

import java.util.\*;

class outerclass {

int n, total=0;

int[] student = new int[6];

double cgpa;

class innerclass {

void calculate() {

Scanner sc = new Scanner(System.in);

for (int i = 1; i <= 5; i++) {

System.out.println("Enter Subject "+i+" Marks: ");

student[i] = sc.nextInt();

}

for (int i = 1; i <= 5; i++)

total+=student[i];

cgpa=(total/(5.0 \* 100)) \* 10;

System.out.print("CGPA: "+cgpa);

}

}

public static void main(String[] args) {

outerclass o = new outerclass();

outerclass.innerclass n = o.new innerclass();

n.calculate();

}

}

**Output:**

Enter Subject 1 Marks: 93

Enter Subject 2 Marks: 87

Enter Subject 3 Marks: 92

Enter Subject 4 Marks: 85

Enter Subject 5 Marks: 91

CGPA: 8.96

**7. Write programs using String and StringBuilder class constructors and methods**

**Code:**

**//StringClass**

class StringClass {

static String str;

StringClass(String s)

{

str=s;

}

public static void main(String[] args) {

StringClass s=new StringClass("Enterprise Java");

System.out.println("Original String: "+str);

System.out.println("Length of String: "+str.length());

System.out.println("Upper Case: "+str.toUpperCase());

System.out.println("Lower case: "+str.toLowerCase());

System.out.println("Character at index 4: "+str.charAt(4));

System.out.println("Substring from index 3: "+str.substring(3));

System.out.println("Substring from index 10 to 15: "+str.substring(10,15));

System.out.println("Concatenated String: "+str.concat(" is easy"));

}

}

**Code:**

**//StringBuilderClass**

class StringBuilderClass {

public static void main(String[] args) {

StringBuilder s = new StringBuilder("Hello");

System.out.println("Original String: " + s);

System.out.println("Length of String: " + s.length());

System.out.println("After Append: " + s.append(" World"));

System.out.println("After insert: " + s.insert(5, " Code"));

System.out.println("After Delete: " + s.delete(6, 11));

System.out.println("After Reverse: " + s.reverse());

System.out.println("Capacity: " + s.capacity());

System.out.println("Character at Index 4: " + s.charAt(4));

}

}

**Output:**

**//StringClass**

Original String: Enterprise Java

Length of String: 15

Upper Case: ENTERPRISE JAVA

Lower case: enterprise java

Character at index 4: r

Substring from index 3: erprise Java

Substring from index 10 to 15: Java

Concated String: Enterprise Java is easy

**Output:**

**//StringBuilderClass**

Original String: Hello

Length of String: 5

After Append: Hello World

After insert: Hello Code World

After Delete: Hello World

After Reverse: dlroW olleH

Capacity: 21

Character at Index 4: W

**ASSIGNMENT-3**

**1. Design a BankAccount class with attributes like account number, balance, and methods for deposit, withdrawal, and balance inquiry. Create subclasses SavingsAccount and CheckingAccount with specific features (e.g., interest calculations for savings, overdraft protection for checking). Utilize inheritance to share common code and polymorphism to handle different account types dynamically. Implement exception handling for potential errors (e.g., insufficient funds for withdrawal).**

import java.util.\*;

import java.io.\*;

class BankAccount {

int account\_number;

int balance;

BankAccount(int acc\_no, int bal) {

account\_number = acc\_no;

balance = bal;

}

void deposite() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Deposite Amount:");

int deposite = sc.nextInt();

balance += deposite;

balance\_enquiry();

}

void withdraw() throws insufficientException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Withdrawal Amount:");

int withdraw = sc.nextInt();

if (balance < withdraw)

throw new insufficientException("Insufficient balance");

else

balance -= withdraw;

balance\_enquiry();

}

void balance\_enquiry() {

System.out.println("Balance is:" + balance);

}

}

class SavingsAccount extends BankAccount {

SavingsAccount(int acc\_no, int bal)

{

super(acc\_no,bal);

}

void calculateInterest() {

double interest = balance \* 0.08;

System.out.println("Interest is:" + interest);

}

void overdraft() {

System.out.println("Overdraft Method");

}

}

class CheckingAccount extends BankAccount {

CheckingAccount(int acc\_no, int bal)

{

super(acc\_no,bal);

}

void overdraft() {

System.out.println("Overdraft Checking");

}

}

class insufficientException extends Exception {

public insufficientException(String message) {

super(message);

}

}

class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Account Number:");

int account\_number = sc.nextInt();

System.out.println("Enter the Initial Balance:");

int balance = sc.nextInt();

SavingsAccount s=new SavingsAccount(account\_number, balance);

CheckingAccount c=new CheckingAccount(account\_number, balance);

try {

s.withdraw();

s.calculateInterest();

s.overdraft();

c.overdraft();

} catch (insufficientException e) {

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

}

}

}

**Output:1**

Enter the Account Number:

12345

Enter the Initial Balance:

5000

Enter the Withdrawal Amount:

500

Balance is:4500

Interest is:360.0

Overdraft Method

Overdraft Checking

**Output2:**

Enter the Account Number:

123

Enter the Initial Balance:

1000

Enter the Withdrawal Amount:

2000

Insufficient balance

**2. Create an abstract class Shape with abstract methods for calculating area and perimeter. Implement concrete subclasses Rectangle, Circle, and Triangle that inherit from Shape and provide specific formulas for their respective calculations. Demonstrate the use of abstract classes to define a contract for subclasses and enforce encapsulation.**

**Code:**

import java.util.\*;

import java.io.\*;

abstract class Shape {

public abstract double calculateArea();

public abstract double calculatePerimeter();

}

class Rectangle extends Shape {

double length;

double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

public double calculateArea() {

return length \* width;

}

public double calculatePerimeter() {

return 2 \* (length + width);

}

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double calculateArea() {

return Math.PI \* radius \* radius;

}

public double calculatePerimeter() {

return 2 \* Math.PI \* radius;

}

}

class Triangle extends Shape {

private double side1;

private double side2;

private double side3;

public Triangle(double side1, double side2, double side3) {

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

public double calculateArea() {

double s = (side1 + side2 + side3) / 2;

return Math.sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

}

public double calculatePerimeter() {

return side1 + side2 + side3;

}

}

public class Program {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter length and width of the Rectangle:");

double length = sc.nextDouble();

double width = sc.nextDouble();

Rectangle rectangle = new Rectangle(length,width);

System.out.println("Rectangle Area: " + rectangle.calculateArea());

System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

System.out.println("\nEnter radius of the circle:");

double radius = sc.nextDouble();

Circle circle = new Circle(radius);

System.out.println("Circle Area: " + circle.calculateArea());

System.out.println("Circle Perimeter: " + circle.calculatePerimeter());

System.out.println("\nEnter lengths of the three sides of the triangle:");

double side1 = sc.nextDouble();

double side2 = sc.nextDouble();

double side3 = sc.nextDouble();

Triangle triangle = new Triangle(side1,side2,side3);

System.out.println("Triangle Area: " + triangle.calculateArea());

System.out.println("Triangle Perimeter: " + triangle.calculatePerimeter());

}

}

**Output:1**

Enter length and width of the rectangle:

12

14

Rectangle Area: 168.0

Rectangle Perimeter: 52.0

Enter radius of the circle:

15

Circle Area: 706.8583470577034

Circle Perimeter: 94.24777960769379

Enter lengths of the three sides of the triangle:

10

12

15

Triangle Area: 59.81168364124187

Triangle Perimeter: 37.0

**3. Design a program that deliberately generates different types of exceptions (e.g., ArithmeticException, NullPointerException, indexOutOfBoundsException). Implement try-catch blocks to handle each exception type gracefully, providing informative error messages and preventing program crashes. Explore the use of finally blocks to ensure essential code like closing resources is executed regardless of exceptions.**

**Code:**

import java.util.\*;

import java.io.\*;

class HelloWorld {

public static void main(String[] args) {

try

{

int x=10/0;

}

catch (ArithmeticException e)

{

System.out.println(e);

}

try

{

String s=null;

System.out.println(s.length());

}

catch (NullPointerException e)

{

System.out.println(e);

}

try

{

int a[]={1,2,3,4,5};

a[5]=10;

}

catch (IndexOutOfBoundsException e)

{

System.out.println(e);

}

finally

{

System.out.println("Finally block always executes.");

}

}

}

**Output:**

java.lang.ArithmeticException: / by zero

java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null

java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5

Finally block always executes.

**4. Write a JAVA program which has A Class called Account that creates account with 500Rs minimum balance, a deposit()method to deposit amount, a withdraw() method to withdraw amount and also throwsLessBalanceException if an account holder tries to withdraw money which makes the balance become less than 500Rs. A Class called LessBalanceException which returns the statement that says withdrawamount ( Rs) is not valid. A Class which creates 2 accounts, both account deposit money and one accounttries to withdraw more money which generates a LessBalanceException take appropriate action for the same.**

**Code:**

import java.util.\*;

import java.io.\*;

class Account {

int min\_balance = 500;

int balance;

Account() {

balance = min\_balance;

}

void deposit(int amt) {

balance += amt;

System.out.println("Deposited: " + amt);

}

void withdraw(int amt) throws LessBalanceException {

if (balance - amt < min\_balance) {

throw new LessBalanceException("Withdraw amount "+amt+" is not valid.");

} else {

balance -= amt;

System.out.println("Withdrawn: " + amt);

}

}

}

class LessBalanceException extends Exception {

public LessBalanceException(String s) {

super(s);

}

}

public class Main {

public static void main(String[] args) {

Account a1 = new Account();

Account a2 = new Account();

try {

a1.deposit(2000);

a2.deposit(1000);

a1.withdraw(200);

a2.withdraw(2000);

} catch (LessBalanceException e) {

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

}

System.out.println("Account 1 Balance: " + a1.balance);

System.out.println("Account 2 Balance: " + a2.balance);

}

}

**Output:**

Deposited: 2000

Deposited: 1000

Withdrawn: 200

Withdraw amount 2000 is not valid.

Account 1 Balance: 2300

Account 2 Balance: 1500

(In the Output the minimum balance 500 is added)

**5. Write a JAVA program to implement a Queue using user defined Exception Handling(also make use of throw, throws.).**

**Code:**

import java.util.\*;

import java.io.\*;

class MyQueue {

int MAX\_SIZE = 5;

int rear;

int front;

int size;

int[] elements = new int[MAX\_SIZE];

MyQueue() {

rear = -1;

front = 0;

size = 0;

}

void enqueue(int data) throws FullQueueException {

if (size == MAX\_SIZE) {

throw new FullQueueException("Queue is Full");

} else {

rear = (rear + 1) % MAX\_SIZE;

elements[rear] = data;

System.out.println("Element Inserted " + data);

size++;

}

}

void dequeue() throws EmptyQueueException {

if (size == 0) {

throw new EmptyQueueException("Queue is Empty");

} else {

System.out.println("Element Removed: " + elements[front]);

front = (front + 1) % MAX\_SIZE;

size--;

}

}

}

class EmptyQueueException extends Exception {

EmptyQueueException(String s) {

super(s);

}

}

class FullQueueException extends Exception {

FullQueueException(String s) {

super(s);

}

}

public class Main {

public static void main(String[] args) {

MyQueue q = new MyQueue();

try {

q.enqueue(5);

q.enqueue(6);

q.enqueue(7);

q.enqueue(8);

q.enqueue(9);

q.enqueue(10);

q.dequeue();

q.dequeue();

q.dequeue();

} catch (FullQueueException e) {

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

} catch (EmptyQueueException e) {

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

}

}

}

**Output:**

Element Inserted 5

Element Inserted 6

Element Inserted 7

Element Inserted 8

Element Inserted 9

Queue is Full

**6. Create a user defined exception called EmpCodeException, BasicException and JobException which will be thrown when empnode is not of 5 characters and should consist of atleast one capital letter and minimum of one digit, basic is not between 2000 and 7000 and if job is not “manager” or analyst or clerk with appropriate error message. In an executable class accept ,empcode, ename, job, basic. Validate the input and if invalid data entered, throw the respective exception and accept the respective data once again. Calculate da, hra, net and display the details.**

**Code:**

import java.util.\*;

class Employee {

String empcode, ename, job;

public int basic;

void input\_EmpCode() throws EmpCodeException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter EmpCode:");

empcode = sc.nextLine();

if (empcode.length() != 5 || !empcode.matches(".\*[A-Z].\*") || !empcode.matches(".\*\\d.\*")) {

throw new EmpCodeException("Invalid Employee Code format!");

}

}

void input\_EmpName() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Employee Name:");

ename = sc.nextLine();

}

void input\_EmpJob() throws JobException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Job:");

job = sc.nextLine();

if (!job.equals("manager") && !job.equals("analyst") && !job.equals("clerk")) {

throw new JobException("Invalid Job");

}

}

void input\_EmpBasic() throws BasicException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Basic:");

basic = sc.nextInt();

if (basic < 2000 || basic > 7000) {

throw new BasicException("Basic Salary should be between 2000 and 7000");

}

}

}

class EmpCodeException extends Exception {

EmpCodeException(String s) {

super(s);

}

}

class BasicException extends Exception {

BasicException(String s) {

super(s);

}

}

class JobException extends Exception {

JobException(String s) {

super(s);

}

}

public class Main {

public static void main(String[] args) {

Employee E = new Employee();

boolean validEmpCode = false;

boolean validJob = false;

boolean validBasic = false;

while (!validEmpCode) {

try {

E.input\_EmpCode();

validEmpCode = true; // If no exception, mark as valid

} catch (EmpCodeException e) {

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

}

}

while (!validJob) {

try {

E.input\_EmpJob();

validJob = true; // If no exception, mark as valid

} catch (JobException e) {

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

}

}

while (!validBasic) {

try {

E.input\_EmpBasic();

validBasic = true; // If no exception, mark as valid

} catch (BasicException e) {

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

}

}

E.input\_EmpName();

int da = E.basic \* 30 / 100;

int hra = E.basic \* 20 / 100;

int net = E.basic + da + hra;

System.out.println("Employee Details:");

System.out.println("Employee Code: " + E.empcode);

System.out.println("Employee Name: " + E.ename);

System.out.println("Job: " + E.job);

System.out.println("Basic Salary: " + E.basic);

System.out.println("DA: " + da);

System.out.println("HRA: " + hra);

System.out.println("Net Salary: " + net);

}

}

**Output:1**

Enter EmpCode: Java1

Enter Job: manager

Enter Basic: 6800

Enter Employee Name: Vivian

Employee Details:

Employee Code: Java1

Employee Name: Vivian

Job: manager

Basic Salary: 6800

DA: 2040

HRA: 1360

Net Salary: 10200

**Output:2**

Enter EmpCode:

java

Invalid Employee Code format!

Enter EmpCode:

Java1

Enter Job:

Professor

Invalid Job

Enter Job:

clerk

Enter Basic:

5000

Enter Employee Name:

7000

Employee Details:

Employee Code: Java1

Employee Name: 7000

Job: clerk

Basic Salary: 5000

DA: 1500

HRA: 1000

Net Salary: 7500

**7. Design a class Product with attributes like name, price, and quantity. Create a ShoppingCart class that uses an ArrayList to store Product objects. Implement methods for adding, removing, and modifying items in the cart. Encapsulate product attributes and cart operations to ensure data integrity and controlled access.**

**Code:**

import java.util.\*;

class Product {

String name;

double price;

int quantity;

public Product(String name, double price, int quantity) {

this.name = name;

this.price = price;

this.quantity = quantity;

}

public void display() {

System.out.println("Name: " + name + "\tPrice: Rs." + price + "\tQuantity: " + quantity);

}

}

class ShoppingCart {

private ArrayList<Product> products = new ArrayList<>();

public void addProduct(Product p) {

products.add(p);

System.out.println("Successfully added new product:");

p.display();

}

public void removeProduct(int index) {

if (index >= 0 && index < products.size()) {

Product removedProduct = products.remove(index);

System.out.println("Successfully removed the product:");

removedProduct.display();

} else {

System.out.println("Invalid index. No product removed.");

}

}

public void displayProducts() {

System.out.println("Product List:");

for (int i = 0; i < products.size(); i++) {

System.out.print((i + 1) + "\t");

products.get(i).display();

}

}

}

public class Main {

public static void main(String[] args) {

ShoppingCart cart = new ShoppingCart();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\nMENU:");

System.out.println("1. Add Product\n2. Remove Product\n3. Display Products\n4. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.println("Enter product details:");

scanner.nextLine(); // Consume newline

System.out.print("Name: ");

String name = scanner.nextLine();

System.out.print("Price: ");

double price = scanner.nextDouble();

System.out.print("Quantity: ");

int quantity = scanner.nextInt();

cart.addProduct(new Product(name, price, quantity));

break;

case 2:

cart.displayProducts();

System.out.print("Enter index to remove: ");

int index = scanner.nextInt() - 1;

cart.removeProduct(index);

break;

case 3:

cart.displayProducts();

break;

case 4:

System.exit(0);

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

}

**Output:**

MENU:

1. Add Product

2. Remove Product

3. Display Products

4. Exit

Enter your choice: 1

Enter product details:

Name: Apple

Price: 9999

Quantity: 12

Successfully added new product:

Name: Apple Price: Rs.9999.0 Quantity: 12

MENU:

1. Add Product

2. Remove Product

3. Display Products

4. Exit

Enter your choice: 1

Enter product details:

Name: Iphone

Price: 2222

Quantity: 2

Successfully added new product:

Name: Iphone Price: Rs.2222.0 Quantity: 2

MENU:

1. Add Product

2. Remove Product

3. Display Products

4. Exit

Enter your choice: 2

Product List:

1 Name: Apple Price: Rs.9999.0 Quantity: 12

2 Name: Iphone Price: Rs.2222.0 Quantity: 2

Enter index to remove: 2

Successfully removed the product:

Name: Iphone Price: Rs.2222.0 Quantity: 2

**8. Create a simple graphical user interface (GUI) using Swing that has a button and a text field. Implement an ActionListener to handle button clicks and modify the text in the text field accordingly. Demonstrate the use of event handling to respond to user interactions and update the GUI dynamically.**

**Code:**

**Output:**

**9. Create two classes: Producer and Consumer. The Producer thread will generate and add items to a shared buffer (e.g., an ArrayList). The Consumer thread will remove and process items from the buffer. Implement synchronization using wait(), notify(), or notifyAll() to ensure the producer doesn't add items to a full buffer and the consumer doesn't wait endlessly for an empty buffer.**

**Code:**

**Output:**

**ASSIGNMENT 4**

**1. Design a Java JDBC program for an Employee Management System (EMS) that enables adding new employees, updating existing employee details, and generating employee reports. Consider the following methods :**

**addEmployee(String name, String designation, double salary) : Method to add a new employee to the database, accepting parameters for employee name, designation, and salary.**

**updateEmployee(int id, String name, String designation,double salary) : Method to update an existing employee's details in the database,specifying the employee ID along with the updated name, designation, and salary.**

**generateEmployeeReport() : Method to generate a report of all employees currently**

**stored in the database, displaying their ID, name, designation, and salary.**

**Calculate da, hra and net and add the record into the database while ans=’yes’. Finally**

**display all the records.**

**Accept an employee number and display his/her record. Ask for confirmation to delete the record. If yes, delete the record.**

**Create table:**

CREATE TABLE EMS (

Id INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(50),

Salary INT,

designation VARCHAR(50),

da INT,

hra INT,

netsalary INT

);

**Create Procedure:**

**//AddEmployee Procedure**

DELIMITER //

CREATE PROCEDURE add\_emp (IN name VARCHAR(50), IN designation VARCHAR(50), IN salary INT)

BEGIN

INSERT INTO EMS (name, designation, salary) VALUES (name, designation, salary);

END

//

**//UpadateEmployee Procedure**

DELIMITER //

CREATE PROCEDURE update\_emp (IN empid INT, IN updated\_name VARCHAR(50), IN updated\_designation VARCHAR(50), IN updated\_salary INT)

BEGIN

UPDATE EMS SET name=updated\_name, designation=updated\_designation, salary=updated\_salary WHERE id=empid;

END

//

**Code**

import java.util.\*;

import java.io.\*;

import java.sql.\*;

class EMS {

Connection con;

void addEmployee(String name, String designation, int salary) throws SQLException {

CallableStatement stmt = con.prepareCall("{call add\_emp(?,?,?)}");

Scanner sc = new Scanner(System.in);

System.out.println("Do you want to add the record into the database? (Y/N):");

char ch = sc.next().charAt(0);

if (ch == 'Y' || ch == 'y') {

double da = (8.0 / 100) \* salary;

double hra = (10.0 / 100) \* salary;

double netSalary = salary + da + hra;

PreparedStatement p = con.prepareStatement("INSERT INTO EMS (name, designation, salary, da, hra, netsalary) VALUES (?, ?, ?, ?, ?, ?)");

p.setString(1, name);

p.setString(2, designation);

p.setInt(3, salary);

p.setDouble(4, da);

p.setDouble(5, hra);

p.setDouble(6, netSalary);

p.executeUpdate();

System.out.println("Record added successfully");

} else {

stmt.setString(1, name);

stmt.setString(2, designation);

stmt.setInt(3, salary);

stmt.execute();

System.out.println("Record Inserted");

}

}

void updateEmployee(int id, String name, String designation, int salary) throws SQLException {

CallableStatement stmt = con.prepareCall("{call update\_emp(?,?,?,?)}");

stmt.setInt(1, id);

stmt.setString(2, name);

stmt.setString(3, designation);

stmt.setInt(4, salary);

stmt.execute();

System.out.println("Record Updated");

}

void generateEmployeeReport() {

try {

Statement s = con.createStatement();

ResultSet r = s.executeQuery("SELECT \* FROM EMS");

System.out.println("Employee Report:");

System.out.println("ID\tName\tDesignation\tSalary\tDA\tHRA\tNetSalary");

while(r.next()) {

int id = r.getInt("id");

String name = r.getString("name");

String designation = r.getString("designation");

int salary = r.getInt("salary");

int da = r.getInt("da");

int hra = r.getInt("hra");

int netSalary = r.getInt("netSalary");

System.out.println(id + "\t" + name + "\t" + designation + "\t" + salary + "\t" + da + "\t" + hra + "\t" + netSalary);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

void DeleteEmployee(int empid) {

try {

Scanner sc = new Scanner(System.in);

Statement s = con.createStatement();

ResultSet r = s.executeQuery("SELECT \* FROM EMS where id=" + empid);

System.out.println("Employee Report:");

System.out.println("ID\tName\tDesignation\tSalary\tDA\tHRA\tNetSalary");

if (r.next()) { // Check if the result set contains any rows

int id = r.getInt("id");

String name = r.getString("name");

String designation = r.getString("designation");

int salary = r.getInt("salary");

int da = r.getInt("da");

int hra = r.getInt("hra");

int netSalary = r.getInt("netSalary");

System.out.println(id + "\t" + name + "\t" + designation + "\t" + salary + "\t" + da + "\t" + hra + "\t" + netSalary);

System.out.println("Do you want to delete the record from the database? (Y/N):");

char ch = sc.next().charAt(0);

if (ch == 'Y' || ch == 'y') {

PreparedStatement p = con.prepareStatement("DELETE FROM EMS WHERE id=?");

p.setInt(1, empid);

p.executeUpdate();

System.out.println("Record Deleted successfully");

}

} else {

System.out.println("No employee found with ID: " + empid);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

try {

EMS E = new EMS();

String name, desig;

int id;

int salary;

Scanner sc = new Scanner(System.in);

E.con = DriverManager.getConnection("jdbc:mysql://localhost:3306/MCA", "root", "password");

while (true) {

System.out.println("\nEmployee Operation\n1.Add Employee\n2.Update Employee\n3.Display Employee\n4.Delete\n5.Exit\nEnter your choice:");

int choice = sc.nextInt();

switch (choice) {

case 1:

System.out.println("Enter Employee name:");

sc.nextLine(); // consume newline

name = sc.nextLine();

System.out.println("Enter Designation:");

desig = sc.nextLine();

System.out.println("Enter Employee Salary:");

salary = sc.nextInt();

E.addEmployee(name, desig, salary);

break;

case 2:

System.out.println("Enter Employee Id:");

id = sc.nextInt();

sc.nextLine(); // consume newline

System.out.println("Enter Employee name:");

name = sc.nextLine();

System.out.println("Enter Designation:");

desig = sc.nextLine();

System.out.println("Enter Employee Salary:");

salary = sc.nextInt();

E.updateEmployee(id, name, desig, salary);

break;

case 3:

E.generateEmployeeReport();

break;

case 4:

System.out.println("Enter Employee Id:");

id = sc.nextInt();

E.DeleteEmployee(id);

break;

case 5:

E.con.close();

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

} catch (SQLException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**//I have used Callable Statements you can use other statement also**

**To run the code**

**First open one terminal and connect to mysql Server**

**mysql -u root -p //enter command**

**Password: password //it will ask for password enter password**

**Use MCA //to select the database**

**Run the Queries //above mentioned**

**Open one more terminal and compile Run java file**

**javac EMS.java //to compile**

**Run the export command and and specify the path where that mysql jar file is located**

**(need not execute connect.java prgrm)**

**export CLASSPATH=$CLASSPATH:/home/admin-mca/Desktop/mysql-connector-java-8.0.30.jar**

**(if u have kept mysql-connector-java-8.0.30.jar in desktop)**

**java EMS //to run**

**2. Build a JDBC program to deal with Student database. Utilize PreparedStatement to**

**insert, update and delete student records.**

**Create table:**

create table student(id int primary key,name varchar(30),class varchar(30),total int);

**Code:**

import java.util.\*;

import java.sql.\*;

class Student {

Connection con;

void insert() throws SQLException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Student Id:");

int id = sc.nextInt();

sc.nextLine(); // Consume newline character

System.out.println("Enter Student name:");

String name = sc.nextLine();

System.out.println("Enter Student Class:");

String clas = sc.nextLine();

System.out.println("Enter Student Total Marks:");

int total = sc.nextInt();

PreparedStatement p = con.prepareStatement("INSERT INTO student (id, name, class, total) VALUES (?, ?, ?, ?)");

p.setInt(1, id);

p.setString(2, name);

p.setString(3, clas);

p.setInt(4, total);

p.executeUpdate();

System.out.println("Student Record added successfully");

}

void update() throws SQLException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Student Id:");

int studid = sc.nextInt();

sc.nextLine(); // Consume newline character

System.out.println("Enter Student name:");

String name = sc.nextLine();

System.out.println("Enter Student Class:");

String clas = sc.nextLine();

System.out.println("Enter Student Total Marks:");

int total = sc.nextInt();

PreparedStatement p = con.prepareStatement("UPDATE student SET name=?, class=?, total=? WHERE id=?");

p.setString(1, name);

p.setString(2, clas);

p.setInt(3, total);

p.setInt(4, studid);

int rowsUpdated = p.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Student Record updated successfully");

} else {

System.out.println("No record found with Student ID: " + studid);

}

}

void delete() throws SQLException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Student Id:");

int stdid = sc.nextInt();

PreparedStatement p = con.prepareStatement("DELETE FROM student WHERE id=?");

p.setInt(1, stdid);

int rowsDeleted = p.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Record Deleted successfully");

} else {

System.out.println("No record found with Student ID: " + stdid);

}

}

public static void main(String[] args) {

try {

Student S = new Student();

S.con = DriverManager.getConnection("jdbc:mysql://localhost:3306/MCA", "root", "password");

Scanner sc = new Scanner(System.in);

while (true) {

System.out.println("\nStudent Operation\n1.Insert\n2.Update\n3.Delete\n4.Exit\nEnter your choice:");

int choice = sc.nextInt();

switch (choice) {

case 1:

S.insert();

break;

case 2:

S.update();

break;

case 3:

S.delete();

break;

case 4:

S.con.close();

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

} catch (SQLException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Output:**

**To run the code follow the instruction given in 1st question**

Student Operation

1.Insert

2.Update

3.Delete

4.Exit

Enter your choice:

1

Enter Student Id:

177

Enter Student name:

Vivian

Enter Student Class:

1 MCA

Enter Student Total Marks:

99

Student Record added successfully

Student Operation

1.Insert

2.Update

3.Delete

4.Exit

Enter your choice:

3

Enter Student Id:

177

Record Deleted successfully

**3.Design a Java JDBC program that utilizes a CallableStatement to call a stored**

**Procedure in a MySQL database.The stored procedure, named calculate\_total\_salary , takes an employee ID as input and returns the total salary for that employee as output.**

**I have Used same table EMS for this**

**Create Procedure:**

DELIMITER //

CREATE PROCEDURE calculate\_total\_salary (IN empid INT, OUT sal INT)

BEGIN

SELECT salary INTO sal FROM EMS WHERE id = empid;

END

//

**Code:**

import java.util.\*;

import java.sql.\*;

class Employee {

Connection con;

public static void main(String[] args) {

Employee E = new Employee();

Scanner sc = new Scanner(System.in);

try {

E.con = DriverManager.getConnection("jdbc:mysql://localhost:3306/MCA", "root", "password");

System.out.println("Enter Employee Id:");

int id = sc.nextInt();

CallableStatement stmt = E.con.prepareCall("{call calculate\_total\_salary(?, ?)}");

stmt.setInt(1, id);

stmt.execute();

int salary = stmt.getInt(2);

System.out.println("Total Salary for Employee ID " + id + ": " + salary);

stmt.close();

E.con.close();

} catch (SQLException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Output:**

**To run the code follow the instruction given in 1st question**

Enter Employee Id: 1

Total Salary for Employee ID 1: 333

Enter Employee Id: 2

Total Salary for Employee ID 2: 999

**4. Create a client program that will send a file to the server, which will receive the file and save it. (Use TCP/IP)**

**Code:**

**Server.java**

import java.io.\*;

import java.net.ServerSocket;

import java.net.Socket;

public class Server {

public static void main(String[] args) {

try {

ServerSocket ss = new ServerSocket(1234);

System.out.println("Server waiting for client...");

while (true) {

Socket s = ss.accept();

System.out.println("Client connected.");

// Create streams for receiving data

InputStream is = s.getInputStream();

ObjectInputStream objin = new ObjectInputStream(is);

// Read the file name from the client

String fileName = (String) objin.readObject();

// Create a file output stream to save the received file

FileOutputStream fout = new FileOutputStream("received\_" + fileName);

// Receive the file content and save it to the file

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = is.read(buffer)) != -1) {

fout.write(buffer, 0, bytesRead);

}

// Close streams

fout.close();

objin.close();

is.close();

s.close();

// Display a message indicating file saved successfully

System.out.println("File saved successfully: received\_" + fileName);

// Display the content of the received file

displayFileContents("received\_" + fileName);

}

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

}

//display part is not necessary

private static void displayFileContents(String fileName) {

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

System.out.println("Content of " + fileName + ":");

String line;

while ((line = br.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Client.java**

import java.io.\*;

import java.net.Socket;

public class Client {

public static void main(String[] args) {

try {

Socket s = new Socket("localhost",1234);

// Specify the file to send

File fileToSend = new File("File.txt");

// Create streams for sending data

OutputStream os = s.getOutputStream();

ObjectOutputStream objout = new ObjectOutputStream(os);

// Send the file name and file content to the server

objout.writeObject(fileToSend.getName());

FileInputStream fin = new FileInputStream(fileToSend);

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = fin.read(buffer)) != -1) {

os.write(buffer, 0, bytesRead);

}

// Close streams

fin.close();

objout.close();

os.close();

s.close();

System.out.println("File sent successfully.");

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Output:**

**Create one normal text file File.txt and add some contents to that..**

**1st open a terminal compile and run Server program**

**Then open one more terminal compile and run Client program**

**(Client and Server should be run in different terminal)**

**(base) mca@mca-precsn:~/NNM23MC177$ javac Server.java**

**(base) mca@mca-precsn:~/NNM23MC177$ java Server**

**Server waiting for client...**

**Client connected.**

**File saved successfully: received\_File.txt**

**Content of received\_File.txt:**

**this is the sample file created to read the contents**

**Name:Vivain**

**Class:MCA**

**(base) mca@mca-precsn:~/NNM23MC177$ javac Client.java**

**(base) mca@mca-precsn:~/NNM23MC177$ java Client**

**File sent successfully.**

**5. Create class Student with data elements studno, sname, marks1, marks2, marks3, total,percentage. Create method getStudent() which will accept studno, sname, marka1, marks2, marks3. Have calculate() method which will calculate total and percentage. Have**

**dispStudent() which will display the student details.**

**Code:**

import java.util.\*;

class Stud{

int no, m1, m2, m3, total;

double percentage;

String name;

void getStudent() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Student No:");

no = sc.nextInt();

sc.nextLine(); // Consume newline character

System.out.println("Enter Student name:");

name = sc.nextLine();

System.out.println("Enter Student Marks 1:");

m1 = sc.nextInt();

System.out.println("Enter Student Marks 2:");

m2 = sc.nextInt();

System.out.println("Enter Student Marks 3:");

m3 = sc.nextInt();

System.out.println("Student Record added successfully");

}

void calculate() {

total = m1 + m2 + m3;

percentage = (total / 300.0) \* 100;

System.out.println("Student Record calculated successfully");

}

void dispStudent() {

System.out.println("Student id:" + no);

System.out.println("Student Name:" + name);

System.out.println("Student Marks 1:" + m1);

System.out.println("Student Marks 2:" + m2);

System.out.println("Student Marks 3:" + m3);

System.out.println("Total:" + total);

System.out.println("Percentage:" + percentage);

}

public static void main(String[] args) {

Stud S = new Stud();

Scanner sc = new Scanner(System.in);

while (true) {

System.out.println("\nStudent Operation\n1.Insert\n2.Calculate\n3.Display\n4.Exit\nEnter your choice:");

int choice = sc.nextInt();

switch (choice) {

case 1:

S.getStudent();

break;

case 2:

S.calculate();

break;

case 3:

S.dispStudent();

break;

case 4:

System.exit(0);

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

}

}

**Output:**

Student Operation

1.Insert

2.Calculate

3.Display

4.Exit

Enter your choice:

1

Enter Student No:

123

Enter Student name:

Vivian

Enter Student Marks 1:

99

Enter Student Marks 2:

98

Enter Student Marks 3:

97

Student Record added successfully

Student Operation

1.Insert

2.Calculate

3.Display

4.Exit

Enter your choice:

2

Student Record calculated successfully

Student Operation

1.Insert

2.Calculate

3.Display

4.Exit

Enter your choice:

3

Student id:123

Student Name:Vivian

Student Marks 1:99

Student Marks 2:98

Student Marks 3:97

Total:294

Percentage:98.0

Student Record updated successfully

**6.In the client program create an object of class Student and accept details for it. Send**

**it to a server program where you calculate the total and percentage. Display the student**

**details in the client program. (Use TCP/IP)**

**Code:**

**Student.java:**

import java.io.Serializable;

public class Student implements Serializable {

String name;

int[] marks;

int total;

double percentage;

public Student(String name, int[] marks) {

this.name = name;

this.marks = marks;

}

public String getName() {

return name;

}

public int[] getMarks() {

return marks;

}

public int getTotal() {

return total;

}

public void setTotal(int total) {

this.total = total;

}

public double getPercentage() {

return percentage;

}

public void setPercentage(double percentage) {

this.percentage = percentage;

}

public void setName(String name) {

this.name = name;

}

}

**Student\_Server.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class Student\_Server {

public static void main(String[] args) {

try {

ServerSocket ss = new ServerSocket(1234);

System.out.println("Server waiting for client...");

Socket s = ss.accept();

System.out.println("Client connected.");

ObjectOutputStream objout = new ObjectOutputStream(s.getOutputStream());

ObjectInputStream objin = new ObjectInputStream(s.getInputStream());

Student student = (Student) objin.readObject();

int total = 0;

for (int mark : student.getMarks()) {

total += mark;

}

double percentage = (total / 300.0) \* 100;

student.setTotal(total);

student.setPercentage(percentage);

objout.writeObject(student);

objout.flush();

objout.close();

objin.close();

s.close();

ss.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Student\_Client.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class Student\_Client {

public static void main(String[] args) {

try {

Socket s = new Socket("localhost", 1234);

System.out.println("Connected to server.");

ObjectOutputStream objout = new ObjectOutputStream(s.getOutputStream());

ObjectInputStream objin = new ObjectInputStream(s.getInputStream());

Scanner sc = new Scanner(System.in);

System.out.print("Enter student name: ");

String name = sc.nextLine();

int[] marks = new int[3];

System.out.println("Enter Subject Marks: ");

for (int i = 0; i < 3; i++) {

System.out.print("Subject " + (i + 1) + ": ");

marks[i] = sc.nextInt();

}

Student student = new Student(name, marks);

objout.writeObject(student);

objout.flush();

// Receive updated student details from server

Student updatedStudent = (Student) objin.readObject();

System.out.println("\nStudent Details:");

System.out.println("Name: " + updatedStudent.getName());

System.out.println("Total Marks: " + updatedStudent.getTotal());

System.out.println("Percentage: " + updatedStudent.getPercentage());

objout.close();

objin.close();

s.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Output:**

**First Open a terminal Compile and run Student.java**

adminmca@MCA-Lab1:~$ javac Student.java

adminmca@MCA-Lab1:~$ java Student

**Then Compile and run Student\_Server.java**

adminmca@MCA-Lab1:~$ javac Student\_Server.java

adminmca@MCA-Lab1:~$ java Student\_Server

Server waiting for client...

Client connected.

**Then open one more terminal Compile and run Student\_Client.java**

adminmca@MCA-Lab1:~$ javac Student\_Client.java

adminmca@MCA-Lab1:~$ java Student\_Client

Connected to server.

Enter student name: Vivian

Enter Subject Marks:

Subject 1: 99

Subject 2: 98

Subject 3: 97

Student Details:

Name: Vivian

Total Marks: 294

Percentage: 98.0

**(Client and Server should be run in different terminal)**

**7. Create a simple chat application using UDP protocol.**

**Code:**

**UDPChatServer.java**

import java.io.\*;

import java.net.\*;

public class UDPChatServer {

public static void main(String[] args) {

try {

DatagramSocket socket = new DatagramSocket(9876);

while (true) {

byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String clientMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

InetAddress clientAddress = receivePacket.getAddress();

int clientPort = receivePacket.getPort();

System.out.println("Client (" + clientAddress.getHostAddress() + ":" + clientPort + "): " + clientMessage);

BufferedReader serverReader = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Server: ");

String serverMessage = serverReader.readLine();

byte[] sendData = serverMessage.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);

socket.send(sendPacket);

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

**UDPChatClient.java**

import java.io.\*;

import java.net.\*;

public class UDPChatClient {

public static void main(String[] args) {

try {

DatagramSocket socket = new DatagramSocket();

InetAddress serverAddress = InetAddress.getByName("localhost");

int serverPort = 9876;

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

while (true) {

System.out.print("You: ");

String message = reader.readLine();

byte[] sendData = message.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);

socket.send(sendPacket);

byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String serverMessage = new String(receivePacket.getData(), 0, receivePacket.getLength()).trim();

System.out.println("Server: " + serverMessage);

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Output:**

**First Compile and run UDPChatServer.java**

PS D:\MCA\Programs> javac UDPChatServer.java

PS D:\MCA\Programs> java UDPChatServer

Client (127.0.0.1:52989): Hii Server..

Server: May I know who is this?

Client (127.0.0.1:52989): I am Vivian from 1st MCA.

Server: Okay

Client (127.0.0.1:52989): See you later

Server: Bye

**Then Compile and run UDPChatClient.java**

PS D:\MCA\Programs> javac UDPChatClient.java

PS D:\MCA\Programs> java UDPChatClient

You: Hii Server..

Server: May I know who is this?

You: I am Vivian from 1st MCA.

Server: Okay

You: See you later

Server: Bye

You:

**8. Display the contents of the URL http://172.16.2.10/index.html/. Display the host name as well as the port at which it is listening as well as the protocol used**

**Code:**

import java.io.\*;

import java.net.\*;

class DisplayURl {

public static void main(String[] args) {

try {

URL url = new URL("http://172.16.2.10/index.html/");

URLConnection conn = url.openConnection();

System.out.println("Host name: " + url.getHost());

System.out.println("Port: " + url.getPort());

System.out.println("Protocol: " + url.getProtocol());

// Read the content from the URL

BufferedReader r = new BufferedReader(new InputStreamReader(conn.getInputStream()));

String line;

System.out.println("Content:");

while ((line = r.readLine()) != null) {

System.out.println(line);

}

r.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Sample Output:**

Host name: 172.16.2.10

Port: -1

Protocol: http

Content:

<!DOCTYPE html>

<html>

<head>

<title>Sample Page</title>

</head>

<body>

<h1>Hello, World!</h1>

<p>This is a sample HTML page.</p>

</body>

</html>

**9. Accept empno, ename, basic into local variables on the client side. Pass the basic to remote methods da(), hra(), net() which will receive basic and calculate da. Hra and net and return them to the client. Display empo, ename, basic, da, hra, net on the client side. (Use RMI).**

**Code:**

**SalaryCalculation.java**

import java.rmi.RemoteException;

import java.rmi.Remote;

interface SalaryCalculation extends Remote {

double CalculateDA(double basic) throws RemoteException;

double CalculateHRA(double basic) throws RemoteException;

double CalculateNet(double basic) throws RemoteException;

}

**CalculationImplementation.java**

import java.rmi.server.UnicastRemoteObject;

import java.rmi.RemoteException;

class CalculationImplementation extends UnicastRemoteObject implements SalaryCalculation {

CalculationImplementation() throws RemoteException {

super();

}

public double CalculateDA(double basic) {

return basic \* 0.08;

}

public double CalculateHRA(double basic) {

return basic \* 0.1;

}

public double CalculateNet(double basic) {

return basic + CalculateDA(basic) + CalculateHRA(basic);

}

}

**EmployeeClient .java**

import java.rmi.\*;

import java.util.Scanner;

public class EmployeeClient {

public static void main(String[] args) {

try {

Scanner sc = new Scanner(System.in);

SalaryCalculation stub = (SalaryCalculation) Naming.lookup("rmi://localhost:5001/employee");

System.out.println("Enter EMP no");

String empNo = sc.nextLine();

System.out.println("Enter Emp Name");

String empName = sc.nextLine();

System.out.println("Enter Basic salary");

double basic = sc.nextDouble();

double da, hra, net;

da = stub.CalculateDA(basic);

hra = stub.CalculateHRA(basic);

net = stub.CalculateNet(basic);

System.out.println("Emp No : " + empNo + "\tEmp Name : " + empName + "\tBasic : " + basic);

System.out.println("Da : " + da + "\tHRA : " + hra + "\tNet Salary : " + net);

} catch (Exception e) {

System.out.println(e);

}

}

}

**EmployeeServer.java**

import java.rmi.\*;

public class EmployeeServer {

public static void main(String[] args){

try {

// Start RMI registry on port 5001

java.rmi.registry.LocateRegistry.createRegistry(5001);

// Create remote object

SalaryCalculation stub = new CalculationImplementation();

// Bind the remote object to the registry

Naming.rebind("rmi://localhost:5001/employee", stub);

System.out.println("Server ready");

} catch (Exception e) {

System.out.println(e);

}

}

}

**Steps to Run:**

**1.Compile Java Files:**

javac SalaryCalculation.java

javac CalculationImplementation.java

javac EmployeeClient.java

javac EmployeeServer.java

**2.Start RMI Registry:**

start rmiregistry

**3.Run Server Application:**

PS D:\MCA\Programs\RMI> java EmployeeServer

Server ready

**4.Run Client Application:**

PS D:\MCA\Programs\RMI> java EmployeeClient

Enter EMP no

123

Enter Emp Name

Vivian

Enter Basic salary

12000

Emp No : 123 Emp Name : Vivian Basic : 12000.0

Da : 960.0 HRA : 1200.0 Net Salary : 14160.0

**10. Accept studno, name, marks1, marks2, marks3 into Student class type of object on the client side. Send the entire object to a remote method where you calculate total and percentage for the student object. Display the resultant student object contents on the client side. (Do not have calculate() in Student class, it should only have getStudent() and dispStudent()). Use RMI.**

**Code:**

**Student.java**

import java.io.Serializable;

public class Student implements Serializable {

private int studentNo;

private String name;

private int marks1, marks2, marks3;

private int total;

double percentage;

public Student(int studentNo, String name, int marks1, int marks2, int marks3){

this.studentNo = studentNo;

this.name = name;

this.marks1 = marks1;

this.marks2 = marks2;

this.marks3 = marks3;

}

public int getMarks1(){

return marks1;

}

public int getMarks2(){

return marks2;

}

public int getMarks3(){

return marks3;

}

public void setTotal(int total){

this.total = total;

}

public void setPercentage(double percentage) {

this.percentage = percentage;

}

public void display(){

System.out.println("Student No : " + studentNo + "\nStudent Name : " + name + "\nMarks 1 : " + marks1 + "\nMarks 2 : " + marks2 + "\nMarks 3 : " + marks3 + "\nTotal : " + total + "\nPercentage : " + percentage);

}

}

**StudentService.java**

import java.rmi.\*;

public interface StudentService extends Remote {

void calculate(Student student) throws RemoteException;

}

**StudentServiceImpl.java**

import java.rmi.\*;

import java.rmi.server.\*;

public class StudentServiceImpl extends UnicastRemoteObject implements StudentService {

StudentServiceImpl() throws RemoteException{

super();

}

public void calculate(Student s) throws RemoteException {

int total = s.getMarks1() + s.getMarks2() + s.getMarks3();

double percentage = (double) total / 300 \* 100;

s.setTotal(total);

s.setPercentage(percentage);

s.display();

}

}

**StudentServer.java**

import java.rmi.Naming;

public class StudentServer {

public static void main(String[] args){

try {

// Start RMI registry on port 5001

java.rmi.registry.LocateRegistry.createRegistry(5001);

// Create remote object

StudentService stub = new StudentServiceImpl();

// Bind the remote object to the registry

Naming.rebind("rmi://localhost:5001/student", stub);

System.out.println("Server ready");

} catch (Exception e) {

System.out.println(e);

}

}

}

**StudentClient.java**

import java.rmi.Naming;

import java.util.Scanner;

public class StudentClient {

public static void main(String[] args){

try {

Scanner sc = new Scanner(System.in);

StudentService stub = (StudentService) Naming.lookup("rmi://localhost:5001/student");

System.out.println("Enter student no");

int stNo = sc.nextInt();

sc.nextLine();

System.out.println("Enter student Name");

String name = sc.nextLine();

System.out.println("Enter 3 subject marks ");

int m1 = sc.nextInt();

int m2 = sc.nextInt();

int m3 = sc.nextInt();

Student st = new Student(stNo, name, m1, m2, m3);

stub.calculate(st);

} catch (Exception e) {

System.out.println(e);

}

}

}

**Output:**

**Steps to Run:**

**1.Compile Java Files:**

**2.Start RMI Registry:**

**3.Run Server Application:**

PS D:\MCA\Programs\RMI> javac Student.java

PS D:\MCA\Programs\RMI> javac StudentService.java

PS D:\MCA\Programs\RMI> javac StudentServiceImpl.java

PS D:\MCA\Programs\RMI> javac StudentServer.java

PS D:\MCA\Programs\RMI> start rmiregistry

PS D:\MCA\Programs\RMI> java StudentServer

Server ready

Student No : 177

Student Name : Vivian

Marks 1 : 99

Marks 2 : 98

Marks 3 : 97

Total : 294

Percentage : 98.0

**4.Run Client Application:**

PS D:\MCA\Programs\RMI> javac StudentClient.java

PS D:\MCA\Programs\RMI> java StudentClient

Enter student no

177

Enter student Name

Vivian

Enter 3 subject marks

99

98

97