1. **Write a Java program to Take three numbers from the user and print the greatest number.**

**Code:**

package Module2;

import java.util.\*;

class M2\_1 {

public static void main(String[] args){

int n1, n2, n3;

Scanner sc = new Scanner(System.in);

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

n1 = sc.nextInt();

System.out.print("\nEnter Second Number: ");

n2 = sc.nextInt();

System.out.print("\nEnter Third Number: ");

n3 = sc.nextInt();

if(n1 > n2){

if(n1 > n3){

System.out.println(String.format("%d is greatest number.",n1));

}

else{

System.out.println(String.format("%d is the greatest number.",n3));

}

}

else{

if(n2 > n3){

System.out.println(String.format("%d is greatest number.",n2));

}

else{

System.out.println(String.format("%d is greatest number.",n3));

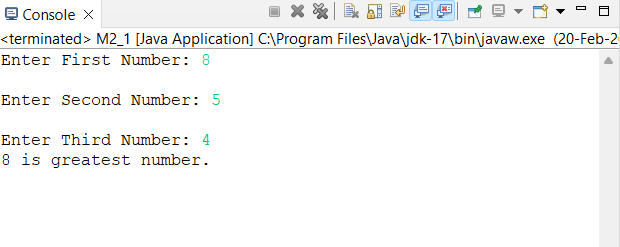
}

}

sc.close();

}

}

**Output:**

1. **Write a Java program that takes the user to provide a single character from the alphabet. Print Vowel or Consonant, depending on the user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.**

**Code:**

package Module2;

import java.util.\*;

class M2\_2{

public static void main(String[] args){

String s;

Scanner sc = new Scanner(System.in);

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

s = sc.next().toLowerCase();

boolean uppercase = s.charAt(0) >= 65 && s.charAt(0) <= 90;

boolean lowercase = s.charAt(0) >= 97 && s.charAt(0) <= 122;

boolean vowels = s.equals("a") || s.equals("e") || s.equals("i") || s.equals("o") || s.equals("u");

if(s.length() > 1) {

System.out.println("Enter only single character!");

}

else if(!(uppercase || lowercase)) {

System.out.println("Not a letter! Enter lowercase or uppercase letter.");

}

else if(vowels) {

System.out.println("'"+s+"' is a vowel.");

}

else {

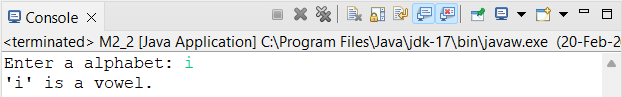
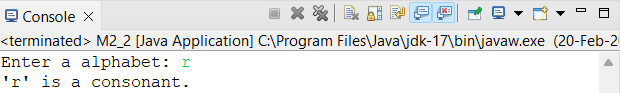
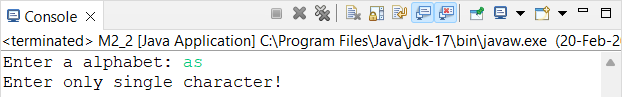
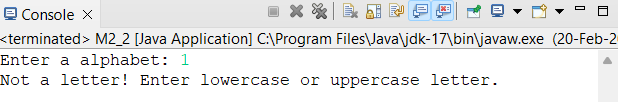
System.out.println("'"+s+"' is a consonant.");

}

sc.close();

}

}



**Output:**

1. **Write a Java program that takes a year from user and print whether that year is a leap year or not.**

**Code:**

package Module2;

import java.util.\*;

class M2\_3{

public static void main(String[] args){

int year;

Scanner sc = new Scanner(System.in);

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

year = sc.nextInt();

if(year % 2 == 0){

System.out.println("Leap year");

}

else{

System.out.println("Non-Leap year");

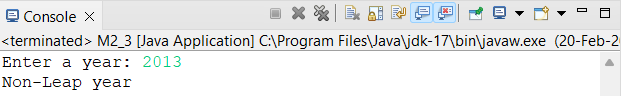
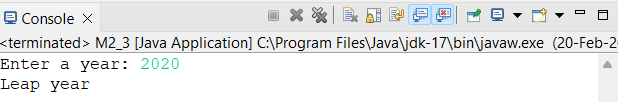
}

sc.close();

}

}

**Output:**



1. **Write a program in Java to display the first 10 natural numbers using while loop.**

**Code:**

package Module2;

class M2\_4{

public static void main(String[] args){

int i = 0;

System.out.println("First ten natural numbers:\n");

while(i <= 10){

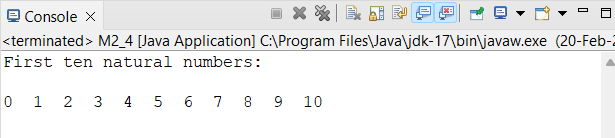
System.out.print(i+" ");

i++;

}

}

}

**Output:**

1. **Write a program in Java to input 5 numbers from keyboard and find their sum and average using for loop.**

**Code:**

package Module2;

import java.util.\*;

class M2\_5{

public static void main(String[] args){

int[] num\_array = new int[5];

int sum = 0, avg;

Scanner sc = new Scanner(System.in);

System.out.print("Enter Any 5 Numbers\n");

for(int i = 0; i < num\_array.length; i++){

System.out.print(String.format("Enter Number %d: ",i+1));

num\_array[i] = sc.nextInt();

}

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

for(int i = 0; i < num\_array.length; i++){

System.out.print(num\_array[i]+" ");

}

for(int i = 0; i < num\_array.length; i++){

sum = sum + i;

}

System.out.println(String.format("\nSum Of Above Numbers: %d",sum));

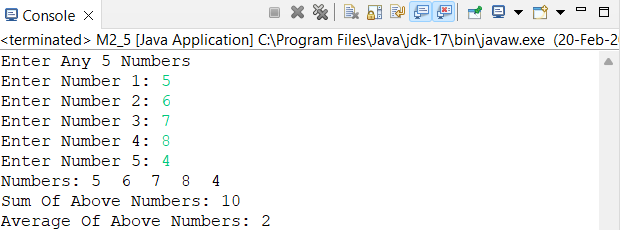
avg = sum / num\_array.length;

System.out.println(String.format("Average Of Above Numbers: %d",avg));

sc.close();

}

}

**Output:**

1. **Write a program in Java to display the pattern like right angle triangle with a number.**

**1**

**12**

**123**

**1234**

**12345**

**Code:**

package Module2;

import java.util.\*;

class M2\_6{

public static void main(String[] args){

int n;

Scanner sc = new Scanner(System.in);

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

n = sc.nextInt();

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

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

System.out.print(j+1+" ");

}

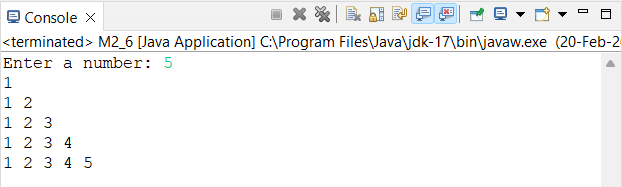
System.out.println();

}

sc.close();

}

}

**Output:**

1. **Write a program in Java to make such a pattern like right angle triangle with number increased by 1 The pattern like:**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

**Code:**

package Module2;

import java.util.\*;

class M2\_7{

public static void main(String[] args){

int n,k=1;

Scanner sc = new Scanner(System.in);

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

n = sc.nextInt();

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

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

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

k++;

}

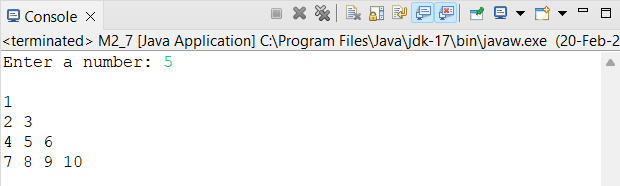
System.out.println();

}

sc.close();

}

}

**Output:**

1. **Write a Java program that reads a positive integer and count the number of digits the number. Input an integer number less than ten billion: 125463 Number of digits in the number: 6**

**Code:**

package Module2;

import java.util.\*;

public class M2\_8 {

public static void main(String[] args) {

int n,count = 0;

Scanner sc = new Scanner(System.in);

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

n = sc.nextInt();

System.out.print("Digits in Number: ");

while(n != 0) {

n = (n - (n % 10)) / 10;

count++;

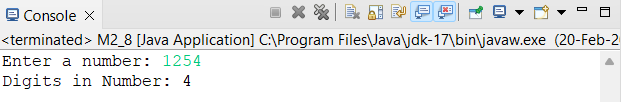
}

System.out.println(count);

sc.close();

}

}

**Output:**

1. **Write a Java program to count the letters, spaces, numbers and other characters of an input string.**

**Code:**

package Module2;

import java.util.\*;

public class M2\_9 {

public static void main(String[] args) {

int digitCount = 0, ltrCount = 0, spsCount = 0, othCount = 0;

String str;

char sps = ' ';

Scanner sc = new Scanner(System.in);

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

str = sc.nextLine();

char[] strArr = new char[str.length()];

strArr = str.toCharArray();

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

if(Character.isDigit(strArr[i])){

digitCount++;

}

else if(Character.isAlphabetic(strArr[i])) {

ltrCount++;

}

else if(strArr[i] == sps) {

spsCount++;

}

else {

othCount++;

}

}

System.out.println("\nString: "+str);

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

System.out.println("\nLetters In String: "+ltrCount);

System.out.println("Digit In String: "+digitCount);

System.out.println("Spaces In String: "+spsCount);

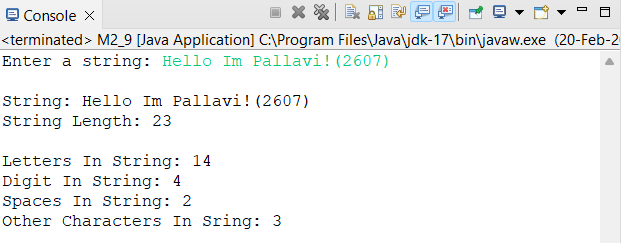
System.out.println("Other Characters In Sring: "+othCount);

sc.close();

}

}

**Output:**

****

1. **Write a Java program to print the ASCII value of a given character.**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_10 {

public static void main(String[] args) {

char c;

Scanner sc = new Scanner(System.in);

System.out.print("Enter a Character To Know Its ASCII Value: ");

c = sc.next().charAt(0);

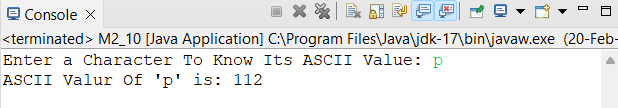
int ascii = (int) c;

System.out.println(String.format("ASCII Valur Of '%s' is: %d",c,ascii));

sc.close();

}

}

**Output:**

1. **Write a Java program that accepts an integer (n) and computes the value of n+nn+nnn. Input number: 5 5 + 55 + 555**

**Code:**

**Output:**

1. **Write a Java program to display the system time.**

**Code:**

package Module2;

//import java.time.LocalTime;

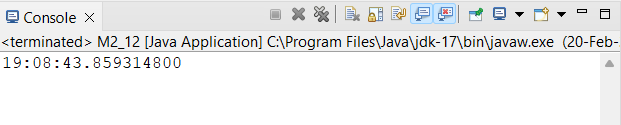
public class M2\_12 {

public static void main(String[] args) {

System.out.println(java.time.LocalTime.now());

}

}

**Output:**

1. **Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_13 {

public static void main(String[] args) {

int[] arrayThree = new int[100];

int[] arrayBoth = new int[100];

int[] arrayFive = new int[101];

Scanner sc = new Scanner(System.in);

System.out.println("Numbers Between 1 To 100 That Are Divisable By 3,5 & Both:");

for(int i = 1; i <= 100; i++) {

if(i % 3 == 0) {

arrayThree[i] = i;

}

if(i % 5 == 0) {

arrayFive[i] = i;

}

if(i % 3 == 0 && i % 5 == 0) {

arrayBoth[i] = i;

}

}

System.out.println("\nValue Divisable by 3: ");

for(int j = 0; j < arrayThree.length; j++) {

if(arrayThree[j] != 0) {

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

}

}

System.out.println("\n\nValue Divisable by 5: ");

for(int j = 0; j < arrayFive.length; j++) {

if(arrayFive[j] != 0) {

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

}

}

System.out.println("\n\nValue Divisable by 3 && 5: ");

for(int j = 0; j < arrayBoth.length; j++) {

if(arrayBoth[j] != 0) {

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

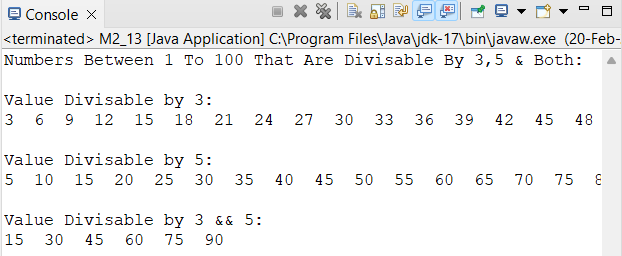
}

}

sc.close();

}

}

**Output:**

1. **W.A.J.P to get the character at the given index within the String. Original String = Tops Technologies! The character at position 0 is T, The character at position 10 is o**

**Code:**

package Module2;

import java.util.\*;

public class M2\_14 {

public static void main(String[] args) {

String str = "Tops Technologies!";

int index;

Scanner sc = new Scanner(System.in);

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

System.out.print("Enter Index:");

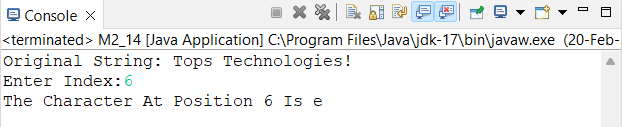
index = sc.nextInt();

System.out.println("The Character At Position "+index+" Is "+str.charAt(index));

sc.close();

}

}

**Output:**

1. **W.A.J.P to concatenate a given string to the end of another string.**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_15 {

public static void main(String[] args) {

String str1, str2;

Scanner sc = new Scanner(System.in);

System.out.print("\nEnter First String: ");

str1 = sc.next();

System.out.print("\nEnter Second String: ");

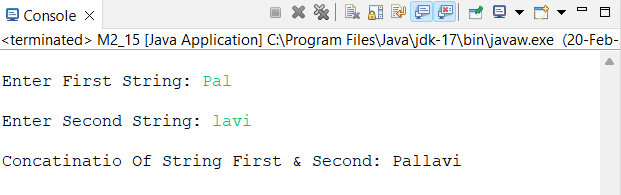
str2 = sc.next();

System.out.print("\nConcatinatio Of String 1 & 2: "+str1.concat(str2));

sc.close();

}

}

**Output:**

1. **W.A.J.P to compare a given string to the specified character sequence. Comparing topsint.com and topsint.com: true Comparing Topsint.com and topsint.com: false**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_16 {

public static void main(String[] args) {

String str1, str2;

Scanner sc = new Scanner(System.in);

System.out.print("Enter First String: ");

str1 = sc.next();

System.out.print("Enter Second String: ");

str2 = sc.next();

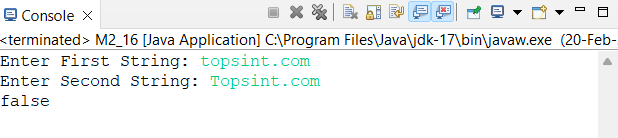
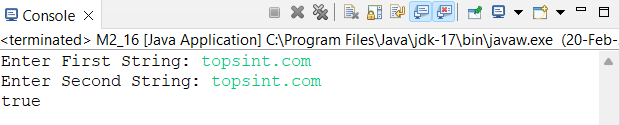
System.out.println(str1.equals(str2));

sc.close();

}

}

**Output:**



1. **W.A.J.P to check whether a given string ends with the contents of another string. "Java Exercises" ends with "se"? False "Java Exercise" ends with "se"? True**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_17 {

public static void main(String[] args) {

String str;

Scanner sc = new Scanner(System.in);

System.out.print("Enter String: ");

str = sc.nextLine();

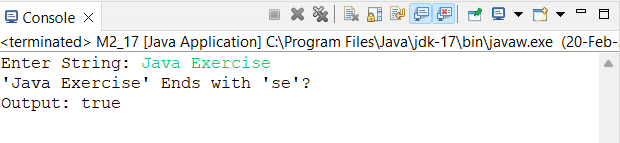
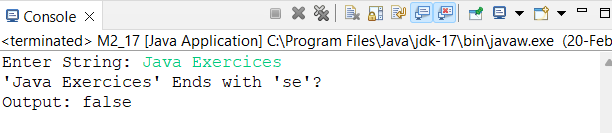
System.out.println("'"+str+"' Ends with 'se'?\nOutput: "+str.endsWith("se"));

sc.close();

}

}

**Output:**



1. **W.A.J.P to check whether a given string starts with the contents of another string. Red is favourite colour. Starts with Red? True Orange is also my favourite colour. Starts with Red? False**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_18 {

public static void main(String[] args) {

String compWith, str;

Scanner sc = new Scanner(System.in);

System.out.print("Enter String: ");

str = sc.nextLine();

System.out.print("Starts With: ");

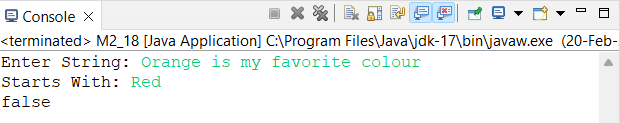
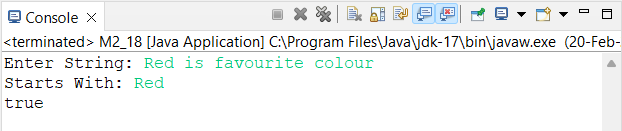
compWith = sc.nextLine();

System.out.println(str.startsWith(compWith));

}

}

**Output:**



1. **W.A.J.P to find all interleaving of given strings. The given strings are: WX YZ The interleaving strings are: YWZX WYZX YWXZ WXYZ YZWX WYXZ**

**Code:**

**Output:**

1. **W.A.J.P to find the second most frequent character in a given string. The given string is: successes The second most frequent char in the string is: c**

**Code:**

**Output:**

1. **Create a class named 'Print Number' to print various numbers of different data types by creating different methods with the same name 'printn' having a parameter for each data type.**

**Code:**

package Module2;

public class M2\_21 {

public static void main(String[] args) {

PrintNumber p = new PrintNumber();

p.printn(12);

p.printn(1.2135);

p.printn(1.0);

p.printn("Pallavi");

}

}

class PrintNumber{

public int printn(int i) {

System.out.println("Integer Data Type: "+i);

return i;

}

public float printn(float f) {

System.out.println("Float Data Type: "+f);

return f;

}

public double printn(double d) {

System.out.println("Double Data Type: "+d);

return d;

}

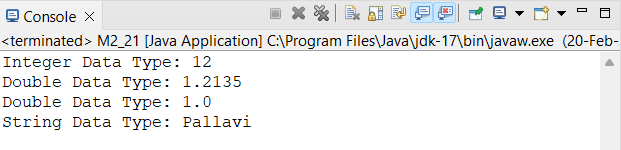
public String printn(String s) {

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

return s;

}

}

**Output:**

1. **Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).**

**Code:**

package Module2;

class intChar{

public void intCharMethod(int n, char c) {

System.out.println("Integer: "+n+" Character: "+c);

}

public void intCharMethod(char c, int n) {

System.out.println("Character: "+c+" Integer: "+n);

}

}

public class M2\_22 {

public static void main(String[] args) {

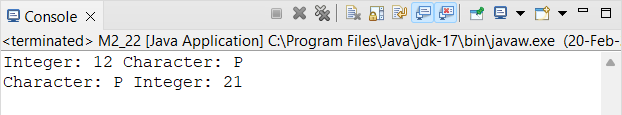
intChar obj = new intChar();

obj.intCharMethod(12, 'P');

obj.intCharMethod('P', 21);

}

}

**Output:**

1. **Create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of a rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square.**

**Code:**

package Module2;

class Area{

public float areaMethod(float length, float breadth) {

return length \* breadth;

}

public float areaMethod(float side) {

return side \* side;

}

}

public class M2\_23 {

public static void main(String[] args) {

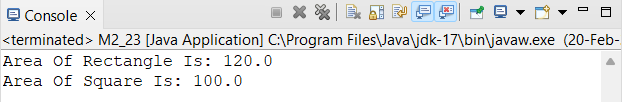
Area A = new Area();

System.out.println("Area Of Rectangle Is: "+A.areaMethod(12,10));

System.out.println("Area Of Square Is: "+A.areaMethod(10));

}

}

**Output:**

1. **Create a class with a method that prints "This is a parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class**

**Code:**

package Module2;

class Parent{

public void parentMethod() {

System.out.println("This is Parent Class");

}

}

class Child extends Parent{

public void childMethod() {

System.out.println("This is Child Class");

}

}

public class M2\_24 {

public static void main(String[] args) {

Parent P = new Parent();

Child C = new Child();

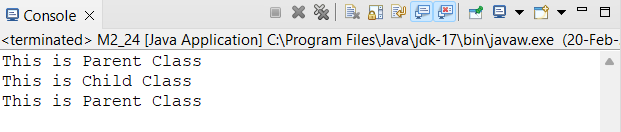
P.parentMethod();

C.childMethod();

C.parentMethod();

}

}

**Output:**

1. **Create a class named 'Member' having the following members: 1. Data members 2. Name 3. Age 4. Phone number 5. Address 6. Salary It also has a method named 'printSalary' which prints the salary of the members.**

**Code:**

package Module2;

class Member{

int menber = 1;

String name = "Pal";

int age = 21;

long phone = 1253645125;

String add = "E2, Dublin Street";

double salary = 30000.00;

public void printSalary() {

System.out.println("Member Salary Is "+salary);

}

}

public class M2\_25 {

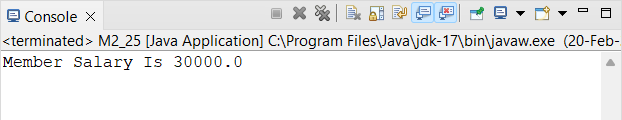
public static void main(String[] args) {

Member M = new Member();

M.printSalary();

}

}

**Output:**

1. **Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize the length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super (s, s)'. Print the area and perimeter of a rectangle and a square.**

**Code:**

package Module2;

class Rectangle{

private float length, breadth;

Rectangle(float length,float breadth){

this.length = length;

this.breadth = breadth;

System.out.println("Length: "+length+"\nBreadth: "+breadth);

}

public void areaMethod() {

System.out.println("Area Of Rectangle: "+length \* breadth);

}

public void perimeterMethod() {

System.out.println("Perimeter Of Rectangle: "+2\*(length + breadth));

}

}

class Square extends Rectangle{

private float side;

public Square(float side){

super(side=10, side=10);

this.side = side;

System.out.println("Side: "+side);

}

public void areaMethod() {

super.areaMethod();

System.out.println("Area Of Square: "+side\*side);

}

public void perimeterMethod() {

super.perimeterMethod();

System.out.println("Perimeter Of Square: "+4\*side);

}

}

public class M2\_26 {

public static void main(String[] args) {

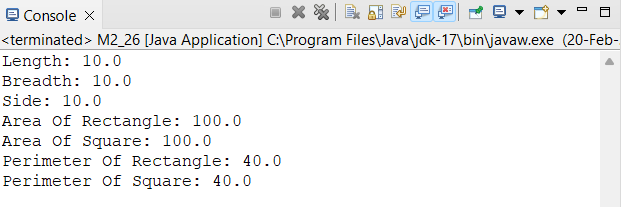
Square S = new Square(10);

S.areaMethod();

S.perimeterMethod();

}

}

**Output:**

1. **Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.**

**Code:**

package Module2;

class Triangle{

float a, b, c, perimeter, p;

double area;

Triangle(){

}

public void getArea() {

p = (a + b + c) / 2;

area = Math.sqrt(p \* (p - a) \* (p - b) \* (p - c));

System.out.println("Area Of Triangle: "+area);

}

public void getPerimeter() {

perimeter = a + b + c;

System.out.println("Perimeter Of Triangle: "+perimeter);

}

}

public class M2\_27 {

public static void main(String[] args) {

Triangle T = new Triangle();

T.a = 3;

T.b = 4;

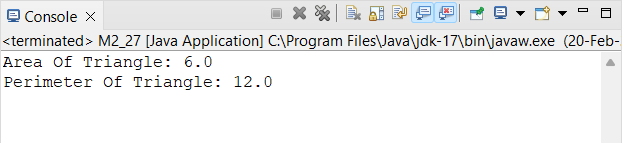
T.c = 5;

T.getArea();

T.getPerimeter();

}

}

**Output:**

1. **Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.**

**Code:**

**Output:**

1. **Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.**

**Code:**

package Module2;

abstract class ParentClass{

public void message() {

}

}

class ChildOne extends Parent{

public void message() {

System.out.println("This is first subclass.");

}

}

class ChildTwo extends Parent{

public void message() {

System.out.println("This is second subclass.");

}

}

public class M2\_29 {

public static void main(String[] args) {

ChildOne C1 = new ChildOne();

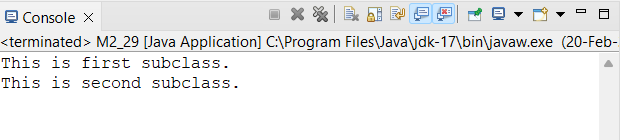
ChildTwo C2 = new ChildTwo();

C1.message();

C2.message();

}

}

**Output:**

1. **Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.**

**Code:**

package Module2;

abstract class Bank{

abstract public void getBalance();

}

class BankA extends Bank{

@Override

public void getBalance() {

System.out.println("Bank A Balance: $"+100);

}

}

class BankB extends Bank{

@Override

public void getBalance() {

System.out.println("Bank B Balance: $"+150);

}

}

class BankC extends Bank{

@Override

public void getBalance() {

System.out.println("Bank C Balance: $"+200);

}

}

public class M2\_30 {

public static void main(String[] args) {

BankA A = new BankA();

BankB B = new BankB();

BankC C = new BankC();

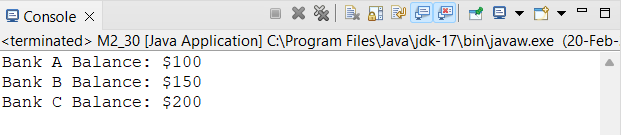
A.getBalance();

B.getBalance();

C.getBalance();

}

}

**Output:**

1. **We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.**

**Code:**

package Module2;

abstract class Marks{

abstract float getPercentage();

}

class A extends Marks{

private float c, java, python;

private float percentage;

A(int c, int java, int python){

this.c = c;

this.java = java;

this.python = python;

System.out.println("Student A Marks: \nC: "+c+"\nJava: "+java+"\nPython: "+python);

}

public float getPercentage() {

percentage = (c + java + python) \* 100 / 300;

return percentage;

}

}

class B extends Marks{

private int php, perl, css, java;

private float percentage;

B(int php, int perl, int css, int java){

this.php = php;

this.perl = perl;

this.css = css;

this.java = java;

System.out.println("\nStudent B Marks: \nPHP: "+php+"\nPerl: "+perl+"\nCSS: "+css+"\nJava: "+java);

}

public float getPercentage() {

percentage = (php + perl + css + java) \* 100 / 400;

return percentage;

}

}

public class M2\_31 {

public static void main(String[] args) {

A a = new A(63, 75, 56);

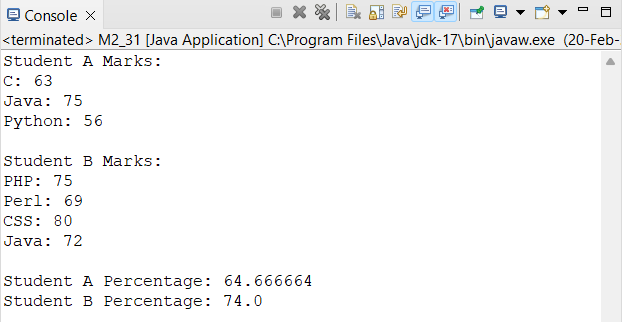
B b = new B(75, 69, 80, 72);

System.out.println("\nStudent A Percentage: "+a.getPercentage());

System.out.println("Student B Percentage: "+b.getPercentage());

}

}

**Output:**

1. **Write a program to print the factorial of a number by defining a method named 'Factorial'. Factorial of any number n is represented by n! And is equal to 1\*2\*3\*. \*(n-1) \*n. E.g.- 4! = 1\*2\*3\*4 = 24 3! = 3\*2\*1 = 6 2! = 2\*1 = 2 Also, 1! = 1 0! = 0**

**Code:**

package Module2;

import java.util.Scanner;

public class M2\_32 {

public static void main(String[] args) {

int n;

long fact = 1;

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number to find it's factorial: ");

n = sc.nextInt();

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

fact = fact \* i;

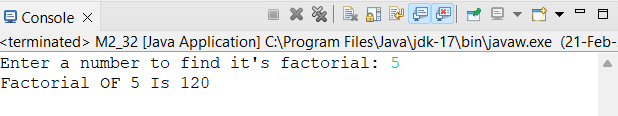
}

System.out.println("Factorial OF "+n+" Is "+fact);

sc.close();

}

}

**Output:**

1. **We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.**

**Code:**

package Module2;

abstract class Shape{

abstract public double RectangleArea(double length, double breadth);

abstract public double SquareArea(double side);

abstract public double CircleArea(double radius);

}

class AreaOfShape extends Shape{

private double pi = 3.14;

@Override

public double RectangleArea(double length, double breadth) {

return length \* breadth;

}

@Override

public double SquareArea(double side) {

return 4\*side;

}

@Override

public double CircleArea(double radius) {

return pi \* radius \* radius;

}

}

public class M2\_33 {

public static void main(String[] args) {

AreaOfShape A = new AreaOfShape();

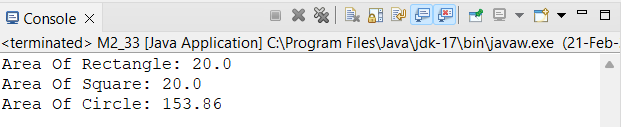
System.out.println("Area Of Rectangle: "+A.RectangleArea(5, 4));

System.out.println("Area Of Square: "+A.SquareArea(5));

System.out.println("Area Of Circle: "+A.CircleArea(7));

}

}

**Output:**

1. **Write a program which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below: Marks Grade 91-100 AA 81-90 AB 71-80 BB 61-70 BC 51-60 CD 41-50 DD 40 Fail**

**Code:**

package Module2;

import java.util.\*;

public class M2\_34 {

public static void main(String[] args) {

int marks;

Scanner sc = new Scanner(System.in);

try {

System.out.print("Enter Your Marks: ");

marks = sc.nextInt();

if(marks >= 91 && marks <= 100) {

System.out.println("Grade: AA");

}

else if(marks >= 81 && marks <= 90) {

System.out.println("Grade: AB");

}

else if(marks >= 71 && marks <= 80) {

System.out.println("Grade: BB");

}

else if(marks >= 61 && marks <= 70) {

System.out.println("Grade: BC");

}

else if(marks >= 51 && marks <= 60) {

System.out.println("Grade: CD");

}

else if(marks >= 41 && marks <= 50) {

System.out.println("Grade: DD");

}

else if(marks < 40) {

System.out.println("Grade: Fail");

}

else {

System.out.println("Invalid Input");

}

}

catch(InputMismatchException e) {

System.out.println("Enter integer value only!");

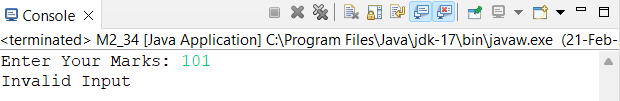
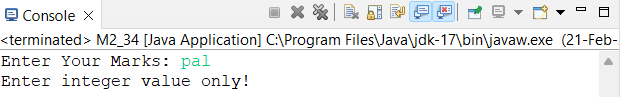
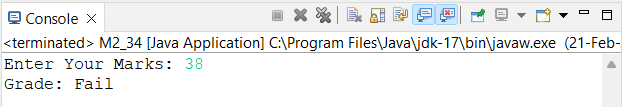
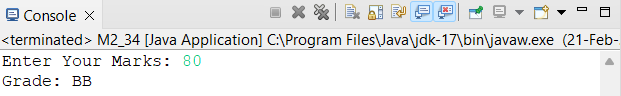
}

sc.close();

}

}

**Output:**



1. **Create a class named 'Shape' with a method to print "This is this is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.**

**Code:**

package Module2;

class ShapeCls{

public void display() {

System.out.println("This is Shape Class.");

}

}

class RectangleCls extends ShapeCls{

public void display() {

super.display();

System.out.println("This is rectangular shape.");

}

}

class CircleCls extends ShapeCls{

public void display() {

System.out.println("This is circular shape.");

}

}

class SquareCls extends RectangleCls{

public void display() {

super.display();

System.out.println("Square is a rectangle.");

}

}

public class M2\_35 {

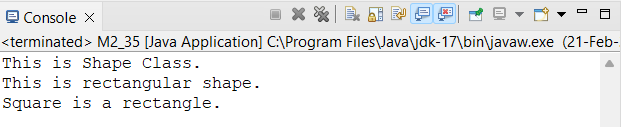
public static void main(String[] args) {

SquareCls S = new SquareCls();

S.display();

}

}

**Output:**

1. **W.A.J. P to demonstrate try catch block,**

**Code:**

**Output:**

1. **Take two numbers from the user and perform the division operation and handle Arithmetic Exception. O/P- Enter two numbers: 10 0 Exception in thread main java.lang.ArithmeticException:/ by zero**
2. **W.A.J. P to demonstrate multiple catch blocks, (one is to handle divide by zero exception and another one is to handle ArrayIndexOutOfBoundException) int a [] =new int [5]; a [5]=30/0;**
3. **W.A.J. P to implement the above program (pro.no-B27) using nesting of try-catch block. try { try {//code} catch (Exception e) {//code} catch (Exception e) {//code}**
4. **W.A.J. P to demonstrate try catch block, take two numbers from the user by Command line argument and perform the division operation and handle Arithmetic O/PException in thread main java. Lang. Arithmetic Exception:/ by zero**
5. **W.A.J.P to create the validate method that takes integer value as a parameter. If the age is less than 18, then throw an Arithmetic Exception otherwise print a message welcome to vote.**

**O/P- Enter your age: 16**

**Exception in thread main java. Lang. Arithmetic Exception: not valid**

1. **W.A.J.P to create a custom exception if Customer withdraw amount which is greater than account balance then program will show custom exception otherwise amount will deduct from account balance. Account balance is: 2000 Enter withdraw amount: 2500 Sorry, insufficient balance, you need more 500 Rs. To perform this transaction.**
2. **W.A.J.P to create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.**
3. **W.A.J. P to create one thread by implementing Runnable interface in Class.**
4. **W.A.J. P to create one thread by extending Thread class in another Class.**
5. **W.A.J.P to create 2 threads and execute that threads by providing sleep time as 2000ms and check the execution.**
6. **W.A.J.P to start the same Thread twice by calling start () method twice. Test ThreadTwice1 t1=new TestThreadTwice1(); t1.start (); t1.start ();**
7. **W.A.J.P to create 2 threads and make one thread as Daemon Thread by using set Daemon () method of Thread class and check whether the thread is set daemon or not by using is Daemon () method. TestDaemonThread2 t1=new TestDaemonThread2(); TestDaemonThread2 t2=new TestDaemonThread2(); t1.start(); t1.setDaemon(true);//will throw exception here t2.start();**
8. **Write a Java program to create a new array list, add some colors (string) and print out the collection.**
9. **Write a Java program to iterate through all elements in an array list.**
10. **Write a Java program to insert an element into the array list at the first position.**
11. **Write a Java program to retrieve an element (at a specified index) from a given array list.**
12. **Write a Java program to update specific array element by given element.**
13. **Write a Java program to remove the third element from an array list.**
14. **Write a Java program to search an element in an array list.**
15. **Write a Java program to sort a given array list.**
16. **Write a Java program to copy one array list into another.**
17. **Write a Java program to shuffle elements in an array list.**
18. **Write a Java program to append the specified element to the end of a hash set.**
19. **Write a Java program to iterate through all elements in a hash list.**
20. **Write a Java program to get the number of elements in a hash set.**
21. **Write a Java program to associate the specified value with the specified key in a Hash Map.**
22. **Write a Java program to count the number of key-value (size) mappings in a map.**
23. **Write a Java program to reverse elements in an array list.**
24. **Write a Java program to extract a portion of an array list.**
25. **Write a Java program to compare two array lists.**
26. **Write a Java program of swap two elements in an array list.**
27. **Write a Java program to join two array lists.**
28. **Write a Java program to convert a hash set to an array.**
29. **Write a Java program to convert a hash set to a List/Array List.**
30. **Write a Java program to check whether a map contains key-value mappings (empty) or not.**
31. **Write a Java program to increase the size of an array list.**
32. **Write a Java program to replace the second element of an Array List with the specified element.**
33. **Write a Java program to print all the elements of an Array List using the position of the elements.**
34. **Write a Java program to compare two sets and retain elements which are same on both sets.**
35. **Write a Java program to get a collection view of the values contained in this map.**