**List of Practical :**

**1. Basic datatype and looping**

**a. Write a program for swapping and find a factorial value. Perform swapping without using a third variable.**

public class Main

{

public static void main(String[] args)

{

int a = 5;

int b = 10;

System.out.println("Before swapping, a = " + a + " and b = " + b);

a = a + b;

b = a - b;

a = a - b;

System.out.println("After swapping, a = " + a + " and b = " + b);

int fact=1;

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

{

fact=fact\*i;

}

System.out.println("Factorial of a value " + a +" is: " +fact);

fact=1;

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

{

fact=fact\*i;

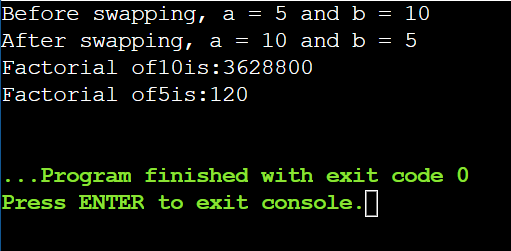
}

System.out.println("Factorial of a value " + b +" is: " +fact);

}

}

**Output:**

****

**b. Write a Java program to check whether a number is even or odd.**

public class Main

{

public static void main(String[] args) {

int number = 29;

if (number % 2 == 0)

System.out.println(number + " is Even");

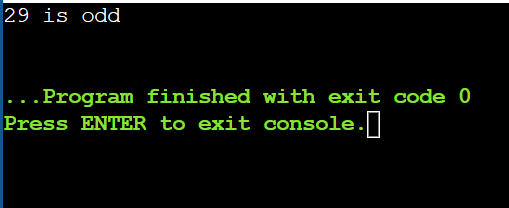
else

System.out.println(number + " is odd");

}

}

**Output:**

****

**c. Write a Java program to check whether a number is prime or not.**

public class Main

{

public static void main(String[] args)

{

int i,m=0,flag=0;

int n=3;

m=n/2;

if(n==0||n==1)

{

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

}

else{

for(i=2;i<=m;i++){

if(n%i==0){

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

flag=1;

break;

}

}

if(flag==0)

{

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

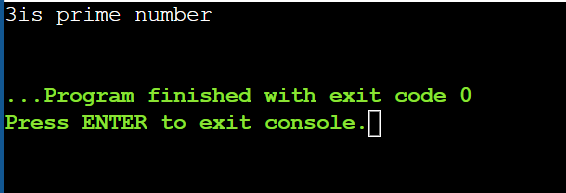
}

}//end of else

}

}

**Output:**

****

**d. Write a Java program to print a Fibonacci series with user input.**

public class Main

{

public static void main(String[] args)

{

int n = 10, firstTerm = 0, secondTerm = 1;

System.out.println("Fibonacci Series till " + n + " terms:");

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

{

System.out.print(firstTerm + ", ");

// compute the next term

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

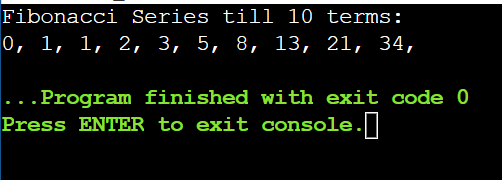
secondTerm = nextTerm;

}

}

}

**Output:**

****

**e. Write a Java Program to Check the Armstrong Number.**

public class Main

{

public static void main(String[] args)

{

int num = 370, number, temp, total = 0;

number = num;

while (number != 0)

{

temp = number % 10;

total = total + temp\*temp\*temp;

number /= 10;

}

if(total == num)

System.out.println(num + " is an Armstrong number");

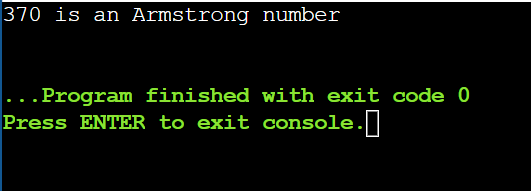
else

System.out.println(num + " is not an Armstrong number");

}

}

**Output:**

****

**f. Swapping Two Numbers in Java Using Third Variable.**

public class Main

{

public static void main(String[] args)

{

int a, b;

a = 10;

b = 20;

System.out.println("Before swapping:");

System.out.println("a = " + a);

System.out.println("b = " + b);

int c;

c = a;

a = b;

b = c;

System.out.println("After swapping:");

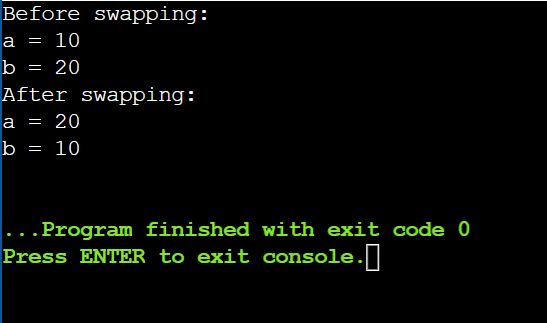
System.out.println("a = " + a);

System.out.println("b = " + b);

}

}

**Output:**

****

**2. Looping and Control structure**

**a. Write a program to accept a number from the user through the command line and display whether the given number is palindrome or not.**

public class Main

{

public static void main(String[] args)

{

int r,sum=0,temp;

int n=1545;

temp=n;

while(n>0){

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

System.out.println(sum + " is palindrome number ");

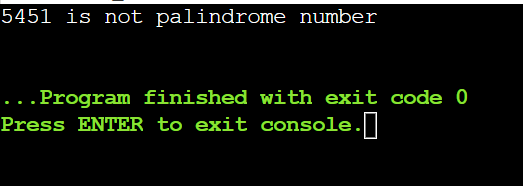
else

System.out.println(sum + " is not palindrome number ");

}

}

**Output:**

****

**b. Java Pattern Programs:**

**Number triangle Pattern**

**1**

**2 2**

**3 3 3**

**4 4 4 4**

**5 5 5 5 5**

**6 6 6 6 6 6**

**Right Half Pyramid Pattern**

**\***

**\*\***

**\*\*\***

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

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

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

**Left Half Pyramid Pattern**

**\***

**\*\***

**\*\*\***

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

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

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

**Reverse Right Half Pyramid Pattern**

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

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

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

**\*\*\***

**\*\***

**\***

**Triangle Star Pattern**

**\***

**\* \***

**\* \* \***

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

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

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

public class Main

{

public static void main(String[] args)

{

int i, j;

int n = 6;

System.out.println("Number triangle Pattern : ");

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

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

System.out.print(" ");

}

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

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

}

System.out.println();

}

System.out.println(" ");

System.out.println("Right Half Pyramid Pattern : ");

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

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

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

}

System.out.println();

}

System.out.println(" ");

System.out.println("Left Half Pyramid Pattern : ");

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

for(j=n;j>=i;j--) {

System.out.print(" ");

}

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

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

}

System.out.println();

}

System.out.println(" ");

System.out.println("Reverse Right Half Pyramid Pattern : ");

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

for (j = n; j >= i; j--) {

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

}

System.out.println();

}

System.out.println(" ");

System.out.println("Triangle Star Pattern : ");

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

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

System.out.print(" ");

}

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

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

}

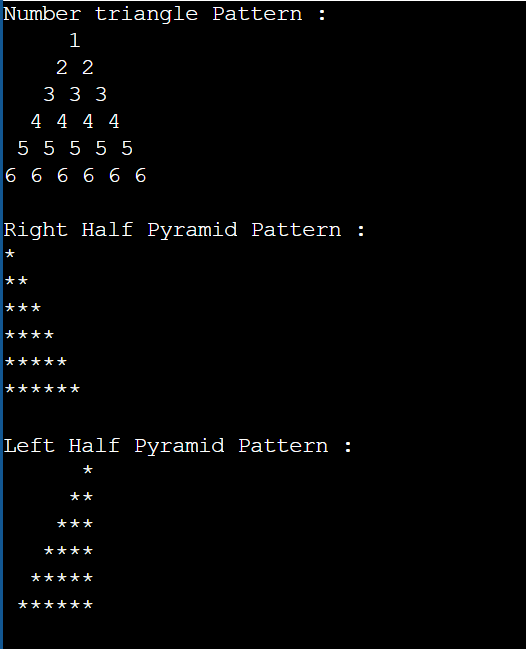
System.out.println();

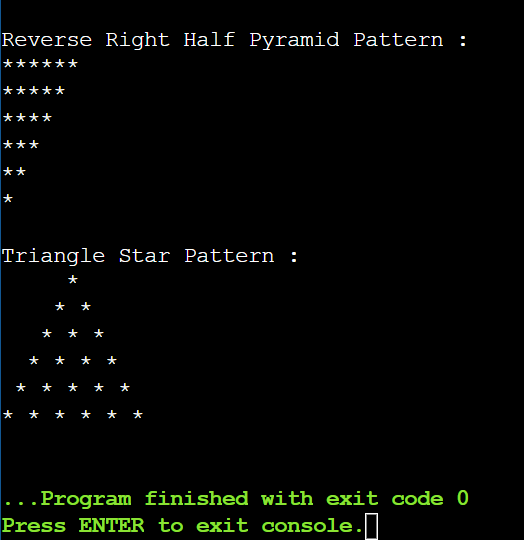
}

}

}

**Output:**

****

****

**3. Array**

**a. Write a program to accept an array of integers from the user through the command line and find prime numbers from the array.**

import java.util.Scanner;

public class Main

{

public static boolean isPrime(int n) {

// Edge cases

if (n <= 1) return false;

if (n <= 3) return true;

// Check if n is divisible by 2 or 3

if (n % 2 == 0 || n % 3 == 0) return false;

// Check the rest of the divisors from 5 to sqrt(n)

for (int i = 5; i \* i <= n; i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return false;

}

// If none of the above conditions are true, n is prime

return true;

}

public static void main(String[] args) {

// Create a scanner object to read input from the user

Scanner sc = new Scanner(System.in);

// Ask the user to enter the size of the array

System.out.println("Enter the size of the array:");

int size = sc.nextInt();

// Declare and initialize an array of integers

int[] arr = new int[size];

// Ask the user to enter the elements of the array

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

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

arr[i] = sc.nextInt();

}

// Close the scanner object

sc.close();

// Print the prime numbers from the array

System.out.println("The prime numbers from the array are:");

for (int num : arr) {

if (isPrime(num)) {

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

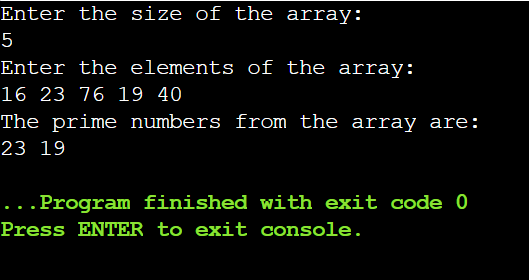
}

}

}

}

**Output:**

****

**4. Class**

**a. Create a class Stack that defines an integer stack that can hold 10 values. Perform push and pop actions in a stack.**

public class Main

{

private int maxSize = 10;

private int top = -1;

private int[] stackArray = new int[maxSize];

public void push(int value) {

if (top < maxSize - 1) {

top++;

stackArray[top] = value;

System.out.println("Pushed: " + value);

} else {

System.out.println("Stack overflow! Cannot push " + value);

}

}

public void pop() {

if (top >= 0) {

int poppedValue = stackArray[top];

top--;

System.out.println("Popped: " + poppedValue);

} else {

System.out.println("Stack is empty! Cannot pop.");

}

}

public void displayStack() {

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

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

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

}

System.out.println();

}

public static void main(String[] args) {

Main stack = new Main();

stack.push(5);

stack.push(10);

stack.push(15);

stack.displayStack();

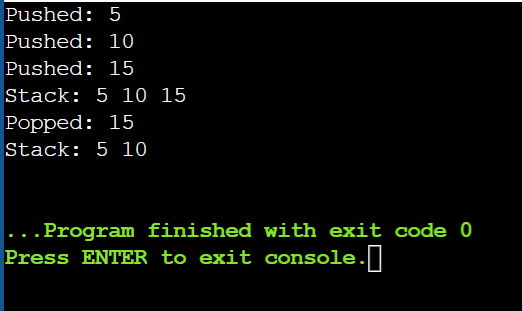
stack.pop();

stack.displayStack();

}

}

**Output:**

****

**b. Write a program to print the area of two rectangles having sides (4,5) and (5,8) respectively by creating a class named 'Rectangle' with a method named 'Area' which returns the area and length and breadth passed as parameters to its constructor.**

class Rectangle

{

private int length;

private int breadth;

public Rectangle(int length, int breadth)

{

this.length = length;

this.breadth = breadth;

}

public int area() {

return length \* breadth;

}

}

public class Main

{

public static void main(String[] args)

{

Rectangle rectangle1 = new Rectangle(4, 5);

Rectangle rectangle2 = new Rectangle(5, 8);

int area1 = rectangle1.area();

int area2 = rectangle2.area();

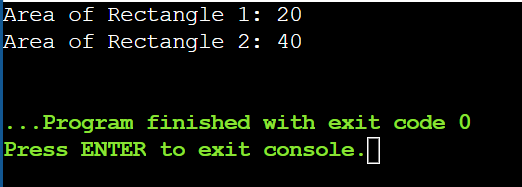
System.out.println("Area of Rectangle 1: " + area1);

System.out.println("Area of Rectangle 2: " + area2);

}

}

**Output:**

****

**c. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.**

import java.util.Scanner;

class Area {

private int length;

private int breadth;

public Area(int length, int breadth) {

this.length = length;

this.breadth = breadth;

}

public int returnArea() {

return length \* breadth;

}

}

public class Main

{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter length of the rectangle: ");

int length = scanner.nextInt();

System.out.print("Enter breadth of the rectangle: ");

int breadth = scanner.nextInt();

Area rectangle = new Area(length, breadth);

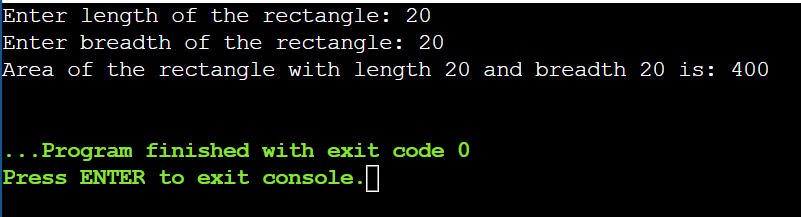
int area = rectangle.returnArea();

System.out.println("Area of the rectangle with length " + length + " and breadth " + breadth + " is: " + area);

}

}

**Output:**

****

**5. Inheritance**

**Write a program to create a class Publisher with attributes publisher name and publisher id. Derive a subclass Book with attributes bookname, bookid and author name. All these data should be entered by the user. Create two methods getdata() and showdata() to display the details of book and publisher.**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.Scanner;

class Publisher {

private String publisherName;

private String publisherId;

public void getData() {

Scanner scanner = new Scanner(System.in);

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

publisherName = scanner.nextLine();

System.out.print("Enter Publisher ID: ");

publisherId = scanner.nextLine();

}

public void showData() {

System.out.println("Publisher Name: " + publisherName);

System.out.println("Publisher ID: " + publisherId);

}

}

class Book extends Publisher {

private String bookName;

private String bookId;

private String authorName;

@Override

public void getData() {

super.getData();

Scanner scanner = new Scanner(System.in);

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

bookName = scanner.nextLine();

System.out.print("Enter Book ID: ");

bookId = scanner.nextLine();

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

authorName = scanner.nextLine();

}

@Override

public void showData() {

super.showData();

System.out.println("Book Name: " + bookName);

System.out.println("Book ID: " + bookId);

System.out.println("Author Name: " + authorName);

}

}

public class Main {

public static void main(String[] args) {

Book book = new Book();

System.out.println("Enter Book and Publisher Details:");

book.getData();

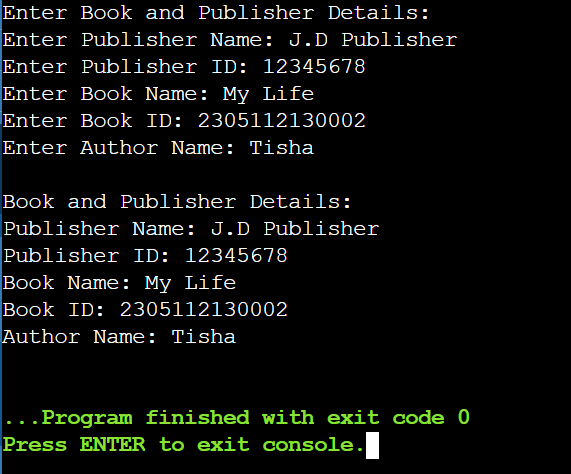
System.out.println("\nBook and Publisher Details:");

book.showData();

}

}

**Output:**

****

**6. Method Overloading**

**a. Write a program to create a class with two methods with same name addfunc(), one accepting two integer parameters and other accepting two double parameters. When method is called, the appropriate method should be selected depending on parameters passed(method overloading).**

class Calculator {

public int addfunc(int a, int b) {

return a + b;

}

public double addfunc(double a, double b) {

return a + b;

}

}

public class Main {

public static void main(String[] args) {

Calculator calculator = new Calculator();

int sumInt = calculator.addfunc(5, 10);

System.out.println("Sum of integers: " + sumInt);

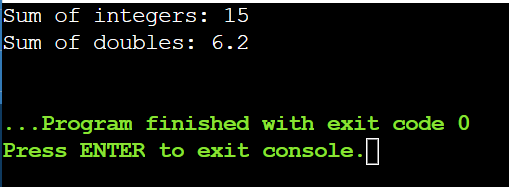
double sumDouble = calculator.addfunc(3.5, 2.7);

System.out.println("Sum of doubles: " + sumDouble);

}

}

**Output:**

****

**7. Super and this keyword**

**Declare a variable called x with integer as the data type in base class and subclass. Make a method named as show() which displays the value of x in the superclass and subclass.**

class BaseClass {

protected int x;

public void setX(int value) {

x = value;

}

public void show() {

System.out.println("Value of x in BaseClass: " + x);

}

}

class SubClass extends BaseClass {

@Override

public void show() {

System.out.println("Value of x in SubClass: " + x);

}

}

public class Main {

public static void main(String[] args) {

BaseClass baseObj = new BaseClass();

baseObj.setX(10);

baseObj.show();

SubClass subObj = new SubClass();

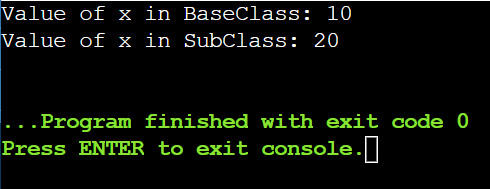
subObj.setX(20);

subObj.show();

}

}

**Output:**

****

**8. Method Overriding**

**a. Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.**

class Shape {

public double getArea() {

System.out.println("Calculating area for generic shape...");

return 0.0;

}

}

class Rectangle extends Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double getArea() {

double area = length \* width;

System.out.println("Calculating area for rectangle: " + area);

return area;

}

}

public class Main {

public static void main(String[] args) {

Rectangle rectangle = new Rectangle(4.0, 5.0);

double rectangleArea = rectangle.getArea();

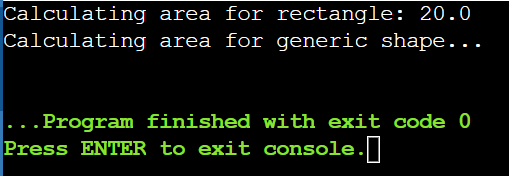
Shape genericShape = new Shape();

double genericShapeArea = genericShape.getArea();

}

}

**Output:**

****

**b. Write a Java program to create a class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().**

class Employee {

public void work() {

System.out.println("Employee is working...");

}

public double getSalary() {

System.out.println("Getting salary for employee...");

return 0.0;

}

}

class HRManager extends Employee {

@Override

public void work() {

System.out.println("HR Manager is managing human resources...");

}

public void addEmployee() {

System.out.println("HR Manager is adding a new employee...");

}

}

public class Main {

public static void main(String[] args) {

Employee employee = new Employee();

employee.work();

double salary = employee.getSalary();

System.out.println("Employee Salary: " + salary);

HRManager hrManager = new HRManager();

hrManager.work();

double hrManagerSalary = hrManager.getSalary();

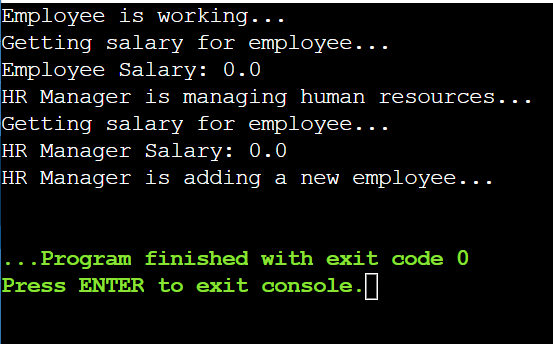
System.out.println("HR Manager Salary: " + hrManagerSalary);

hrManager.addEmployee();

}

}

**Output:**

****

**c. Write a Java program to create a class known as Person with methods called getFirstName() and getLastName(). Create a subclass called Employee that adds a new method named getEmployeeId() and overrides the getLastName() method to include the employee's job title.**

class Person {

private String firstName;

private String lastName;

public Person(String firstName, String lastName) {

this.firstName = firstName;

this.lastName = lastName;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

}

class Employee extends Person {

private String employeeId;

private String jobTitle;

public Employee(String firstName, String lastName, String employeeId, String jobTitle) {

super(firstName, lastName);

this.employeeId = employeeId;

this.jobTitle = jobTitle;

}

public String getEmployeeId() {

return employeeId;

}

@Override

public String getLastName() {

return super.getLastName() + " (" + jobTitle + ")";

}

}

public class Main {

public static void main(String[] args) {

Employee employee = new Employee("John", "Doe", "E12345", "Software Engineer");

System.out.println("First Name: " + employee.getFirstName());

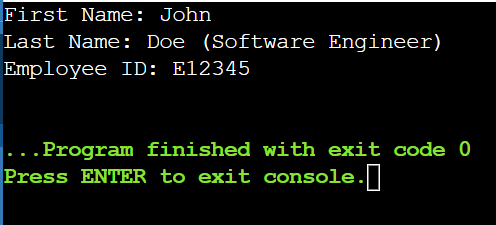
System.out.println("Last Name: " + employee.getLastName());

System.out.println("Employee ID: " + employee.getEmployeeId());

}

}

**Output:**

****

**d. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.**

class Shape {

public double getPerimeter() {

System.out.println("Calculating perimeter for generic shape...");

return 0.0;

}

public double getArea() {

System.out.println("Calculating area for generic shape...");

return 0.0;

}

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double getPerimeter() {

double perimeter = 2 \* Math.PI \* radius;

System.out.println("Calculating perimeter for circle: " + perimeter);

return perimeter;

}

@Override

public double getArea() {

double area = Math.PI \* radius \* radius;

System.out.println("Calculating area for circle: " + area);

return area;

}

}

public class Main {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

double circlePerimeter = circle.getPerimeter();

double circleArea = circle.getArea();

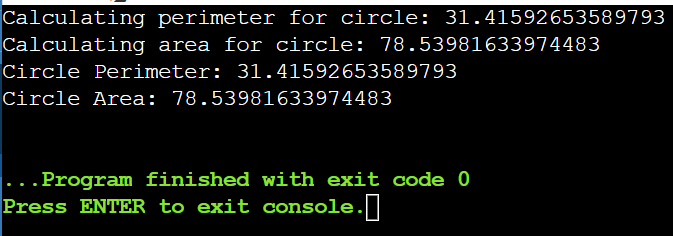
System.out.println("Circle Perimeter: " + circlePerimeter);

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

}

}

**Output:**

****

**9. Final class, abstract class, and interface**

**a. Write a program to calculate the area, circumference and volume for all shapes. [Perform this application using final class, abstract class and interface]**

final class Square {

private double side;

public Square(double side) {

this.side = side;

}

public double calculateArea() {

return side \* side;

}

public double calculatePerimeter() {

return 4 \* side;

}

}

interface ThreeDimensionalShape {

double calculateVolume();

}

abstract class Circle implements ThreeDimensionalShape {

protected double radius;

public Circle(double radius) {

this.radius = radius;

}

public double calculateArea() {

return Math.PI \* radius \* radius;

}

public double calculateCircumference() {

return 2 \* Math.PI \* radius;

}

}

class Sphere extends Circle {

public Sphere(double radius) {

super(radius);

}

@Override

public double calculateVolume() {

return (4.0 / 3.0) \* Math.PI \* Math.pow(radius, 3);

}

}

public class Main {

public static void main(String[] args) {

Square square = new Square(5.0);

System.out.println("Square Area: " + square.calculateArea());

System.out.println("Square Perimeter: " + square.calculatePerimeter());

Sphere sphere = new Sphere(3.0);

System.out.println("Sphere Area: " + sphere.calculateArea());

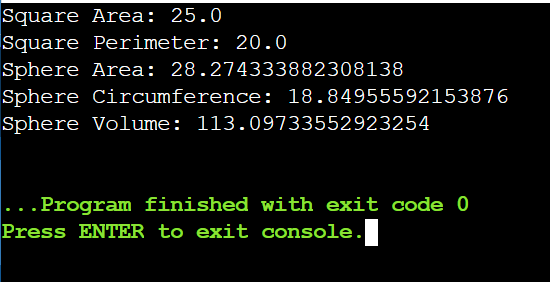
System.out.println("Sphere Circumference: " + sphere.calculateCircumference());

System.out.println("Sphere Volume: " + sphere.calculateVolume());

}

}

**Output:**

****

**b. w.a.p in java to create a class circle to calculate and display the area of the circle to implement Final Variable in Java**

**c. w.a.p in Java to create a class Animal to Implement Final Methods the general characteristics of an Animal along with the constant variables to store the number of legs, ears eyes, and whether the animal has a tail. Create a subclass wolf to display the additional characteristics.**

**d. Write a Java program to create an abstract class BankAccount with abstract methods deposit() and withdraw(). Create subclasses: SavingsAccount and CurrentAccount that extend the BankAccount class and implement the respective methods to handle deposits and withdrawals for each account type.**

**e. Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.**