List of Practical

# Basic datatype and looping

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

Ans)

**package** parctise;

**import** java.util.Scanner;

**public class** swaping {

**public static void** main(String[] args) {

System.***out***.println("Swapping wih three variable : ");

**int** first = 5; **int** second = 10; **int** temp;

System.***out***.println("Number before swapping are : "+first+" " + second); temp = first;

first = second; second = temp;

System.***out***.println("Number after swapping are : "+first+" " + second);

Scanner obj = **new** Scanner(System.***in***); **int** fans = 0; System.***out***.println("Enter a number :"); **int** a =obj.nextInt();

**for**(**int** i = 1; i<=a; i++) { fans += i;

}

System.***out***.println("Factorel of " + a+" is "+fans);

System.***out***.println("Swapping wihout three variable : ");

**int** c=11,b=10;

System.***out***.println("Number before swapping are : "+c+" " + b);

c=c+b; b=c-b;

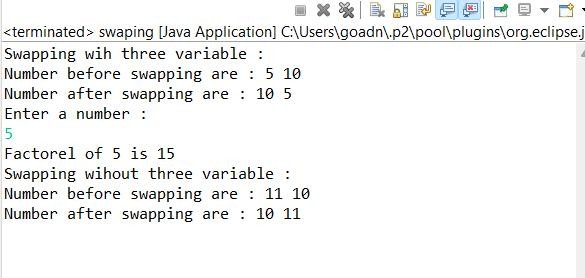
c=c-b;

System.***out***.println("Number after swapping are : "+c+" " + b);

}

}

Output:



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

Ans)

**package** parctise;

**public class** oddeven {

**public static void** main(String[] args) {

**int** number = 2;

**int** number2 = 3;

System.***out***.println("Checking for even number : ");

**if**(number%2 ==0) {

System.***out***.println("Number is even number. "+number);

}**else** {

}

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

System.***out***.println("Checking for odd number : ");

**if**(number2%2 !=0) {

System.***out***.println("Number is odd number. "+number2);

}**else** {

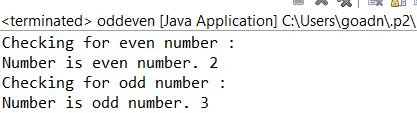
}

System.***out***.println("Number is even number. "+number2);

}

}

Output:



# Write a Java program to check whether a number is prime or not

Ans)

**package** parctise;

**import** java.util.Scanner;

**public class** parctise {

**public static void** main(String[] args) { Scanner sb =**new** Scanner(System.***in***);

**int** con;

**do** {

System.***out***.println("Enter a number : ");

**int** a = sb.nextInt();

**int** fc=0;

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

**int** temp = a;

**if**(temp % i==0) {

fc=fc+1;

}

}

**if**(fc==2) {

System.***out***.println("The number " + a + " is a prime number");

}

## else {

}

System.***out***.println("The number " + a + " is not a prime number");

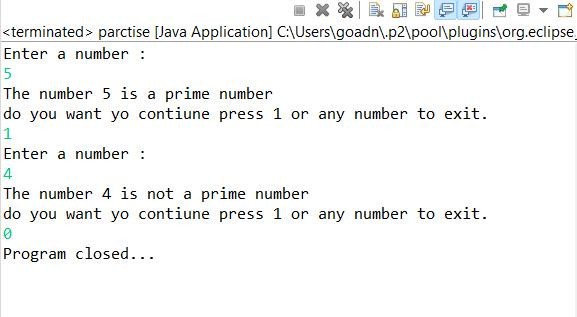
}

}

Output:

System.***out***.println("do you want yo contiune press 1 or any number to exit."); con = sb.nextInt();

}**while**(con == 1); System.***out***.println("Program closed...");



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

Ans)

**package** parctise;

**public class** fibo {

**public static void** main(String[] args) {

**int** a= 5;

**int** b =0,c=1; System.***out***.print(" "+b+" "+c ); **for**(**int** i = 1; i<=a;i++) {

**int** sum = c; sum =b+c; b=c;

c=sum;

System.***out***.print(" "+sum);

}

}

}

Output:



# Write a Java Program to Check the Armstrong Number

Ans)

**package** parctise;

**import** java.lang.Math;

**public class** armstrong {

**public static void** main(String[] args) { String number = "153";

**int** numbera = Integer.*parseInt*(number);

**int** base = number.length();

**int** temp = numbera;

**double** total=0;

**for**(**int** i = 1; i<=base;i++) {

**int** temp2 = temp%10;

total += Math.*pow*(temp2, base); temp = temp/10;

}

**if**(total == numbera) {

System.***out***.println("The number is armstrong number : "+numbera);

}**else** {

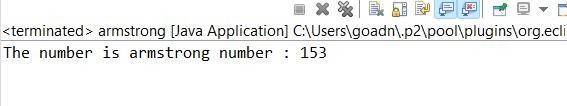
}

System.***out***.println("The number is not armstrong number : "+numbera);

}

}

Output:



# Swapping Two Numbers in Java Using Third Variable

Ans)

**package** parctise;

**public class** swappingwiththree {

**public static void** main(String[] args) { System.***out***.println("Swapping wih three variable : "); **int** first = 5;

**int** second = 10;

**int** temp;

System.***out***.println("Number before swapping are : "+first+" " + second); temp = first;

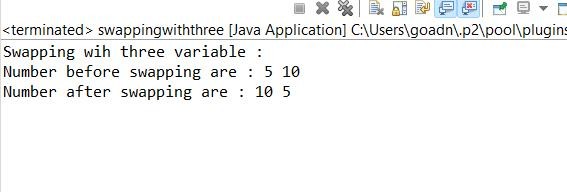
first = second; second = temp;

System.***out***.println("Number after swapping are : "+first+" " + second);

}

}

Output:



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

**Ans)**

**package** parctise;

**public class** palindrome {

**public static void** main(String[] args) {

**int** r,sum=0,temp;

**int** n=15451;

temp=n; **while**(n>0){ r=n%10;

sum=(sum\*10)+r; n=n/10;

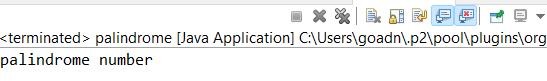
}

**if**(temp==sum) System.***out***.println("palindrome number "); **else**

System.***out***.println("not palindrome");

}

} Output:



# 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

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

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

Ans)

**package** parctise;

**public class** Pattern {

**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("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("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("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("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("\* ");

}

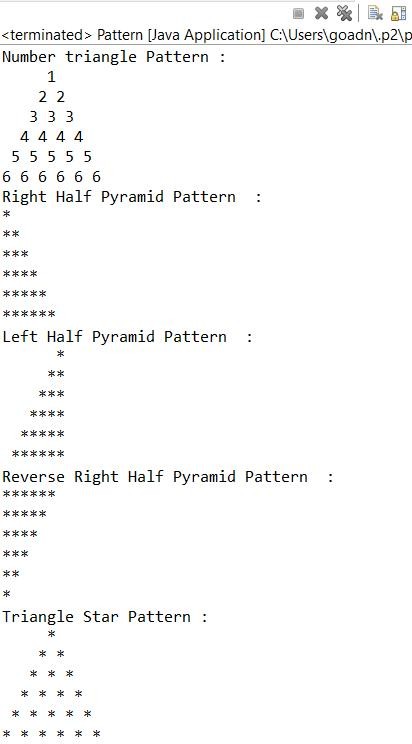
System.***out***.println();

}

}

}

Output:



# Array

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

Ans)

**package** parctise;

**import** java.util.Scanner;

**public class** userinput {

**public static boolean** isPrime(**int** num) {

**if** (num < 2) {

## return false;

}

**for** (**int** i = 2; i < num; i++) {

**if** (num % i == 0) {

## return false;

}

}

## return true;

}

**public static void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter a list of integers separated by spaces : "); String input = scanner.nextLine();

String[] inputNumbers = input.split(" ");

**int**[] numbers = **new int**[inputNumbers.length]; **for** (**int** i = 0; i < inputNumbers.length; i++) {

## try {

numbers[i] = Integer.*parseInt*(inputNumbers[i]);

} **catch** (NumberFormatException e) {

System.***out***.println("Invalid input: " + inputNumbers[i] + " is not a valid integer.");

## return;

}

}

System.***out***.print("Prime numbers in the array: ");

**for** (**int** num : numbers) {

**if** (*isPrime*(num)) { System.***out***.print(num + " ");

}

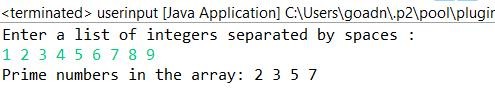
}

System.***out***.println();

}

}

Output:



# Class

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

Ans)

**package** parctise;

**class** stack {

**private int**[] data;

**private int** top;

**public** stack() {

data = **new int**[10]; top = -1;

}

**public void** push(**int** value) {

**if** (top == 9) {

System.***out***.println("Stack is full. Cannot push more elements.");

## return;

}

data[++top] = value; System.***out***.println("Pushed valued : "+value);

}

**public int** pop() {

**if** (top == -1) {

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

**return** -1;

}

**return** data[top--];

}

}

**public class** q4a{

**public static void** main(String[] args) { stack stack = **new** stack();

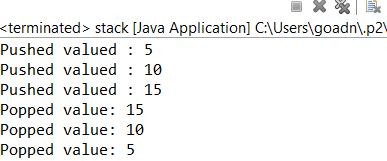
stack.push(5); stack.push(10); stack.push(15);

System.***out***.println("Popped value: " + stack.pop()); System.***out***.println("Popped value: " + stack.pop()); System.***out***.println("Popped value: " + stack.pop());

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**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** q4b {

**public static void** main(String[] args) {

Rectangle rectangle1 = **new** Rectangle(4, 5); Rectangle rectangle2 = **new** Rectangle(5, 8);

**int** area1 = rectangle1.area();

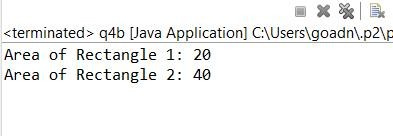
**int** area2 = rectangle2.area();

System.***out***.println("Area of Rectangle 1: " + area1); System.***out***.println("Area of Rectangle 2: " + area2);

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**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** q4c {

**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();

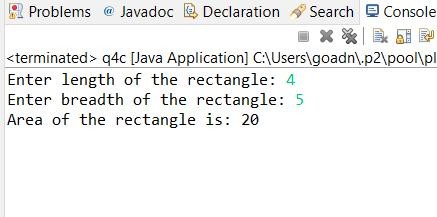
Area rectangle = **new** Area(length, breadth);

**int** area = rectangle.returnArea(); System.***out***.println("Area of the rectangle is: " + area);

}

}

Output:



# Inheritance

Ans)

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.

**package** parctise;

**import** java.util.Scanner;

**class** Publisher {

**private** String publisherName;

**private int** 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.nextInt(); scanner.nextLine(); // Consume the newline character

}

**public void** showData() {

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

}

}

**class** Book **extends** Publisher { **private** String bookName; **private int** bookId; **private** String authorName;

**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.nextInt();

scanner.nextLine(); // Consume the newline character

System.***out***.print("Enter Author Name: "); authorName = scanner.nextLine();

}

**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** q5 {

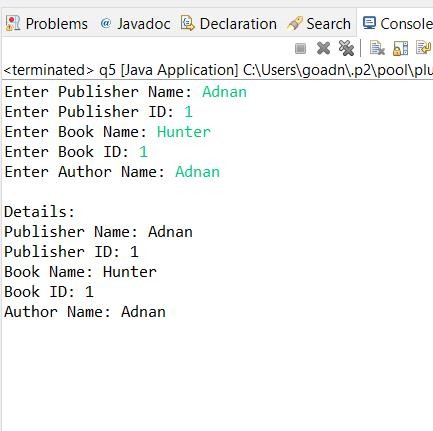
**public static void** main(String[] args) { Book book = **new** Book();

book.getData(); System.***out***.println("\nDetails:"); book.showData();

}

}

Output:



# Method Overloading

Ans)

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).

**package** parctise;

**class** Calculator {

**public int** addFunc(**int** num1, **int** num2) {

**return** num1 + num2;

}

**public double** addFunc(**double** num1, **double** num2) {

**return** num1 + num2;

}

}

**public class** q6 {

**public static void** main(String[] args) { Calculator calculator = **new** Calculator();

**int** sum1 = calculator.addFunc(5, 7);

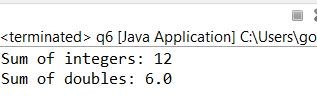
**double** sum2 = calculator.addFunc(3.5, 2.5);

System.***out***.println("Sum of integers: " + sum1); System.***out***.println("Sum of doubles: " + sum2);

}

}

Output:



# 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.

Ans)

**package** parctise;

**class** superclass{

**int** x=5;

**public void** show() {

System.***out***.println("Value of x in superclass is : "+x);

}

}

**class** subclass **extends** superclass{

**int** x;

**public void** show() {

**super**.show();

System.***out***.println("Value of x in superclass from subclass is : "+**super**.x); System.***out***.println("Value of x in subclass is : "+x);

}

}

**public class** q7 {

**public static void** main(String[] args) {

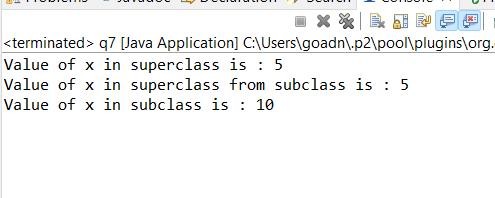
subclass sc1 = **new** subclass(); sc1.x=10;

sc1.show();

}

}

Output:



# Method Overriding

* 1. 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.

Ans)

**package** parctise;

**class** Shape {

**double** getArea(**double** length,**double** width) {

**return** length\*width;

}

}

**class** Rectangle1 **extends** Shape {

@Override

**double** getArea(**double** length,**double** width) {

**return** length \* width;

}

}

**public class** q8a {

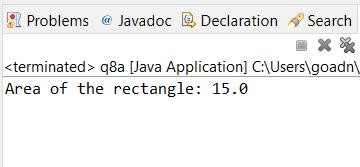
**public static void** main(String[] args) { Rectangle1 rectangle = **new** Rectangle1();

**double** area = rectangle.getArea(5.0, 3.0); System.***out***.println("Area of the rectangle: " + area);

}

}

Output:



* 1. 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().

Ans)

**package** parctise;

**class** Employee {

**int** hours;

**void** work(**int** hours) { System.***out***.println("Employee is working."); **this**.hours=hours;

}

**double** getSalary() {

**return** hours\*100;

}

}

**class** HRManager **extends** Employee {

**int** hours; @Override

**void** work(**int** hours) {

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

**this**.hours=hours; **super**.work(hours);

}

**void** addEmployee() {

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

}

}

**public class** q8b {

**public static void** main(String[] args) { Employee employee = **new** Employee();

employee.work(250);

System.***out***.println("Employee Salary: " + employee.getSalary()); HRManager hrManager = **new** HRManager();

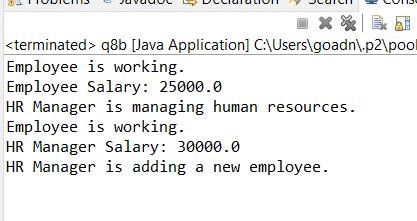
hrManager.work(300);

System.***out***.println("HR Manager Salary: " + hrManager.getSalary()); hrManager.addEmployee();

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**class** Person {

**private** String firstName;

**private** String lastName;

Person(String firstName, String lastName) { **this**.firstName = firstName; **this**.lastName = lastName;

}

String getFirstName() {

**return** firstName;

}

String getLastName() {

**return** lastName;

}

}

**class** Employee1 **extends** Person {

**private int** employeeId;

**private** String jobTitle;

Employee1(String firstName, String lastName, **int** employeeId, String jobTitle) {

**super**(firstName, lastName); **this**.employeeId = employeeId; **this**.jobTitle = jobTitle;

}

@Override

String getLastName() {

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

}

**int** getEmployeeId() {

**return** employeeId;

}

}

**public class** q8c {

**public static void** main(String[] args) {

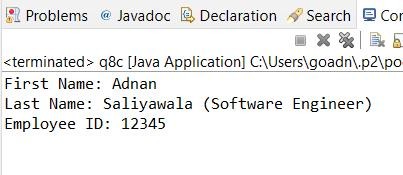
Employee1 employee = **new** Employee1("Adnan", "Saliyawala", 12345, "Software Engineer");

System.***out***.println("First Name: " + employee.getFirstName()); System.***out***.println("Last Name: " + employee.getLastName()); System.***out***.println("Employee ID: " + employee.getEmployeeId());

}

}

Output:



* 1. 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. Ans)

**package** parctise;

**class** Shapes {

**double** getPerimeter() {

**return** 0.0;

}

**double** getArea() {

**return** 0.0;

}

}

**class** Circle **extends** Shapes {

**private double** radius;

Circle(**double** radius) {

**this**.radius = radius;

}

@Override

**double** getPerimeter() {

**return** 2 \* Math.***PI*** \* radius;

}

@Override

**double** getArea() {

**return** Math.***PI*** \* radius \* radius;

}

}

**public class** q8d {

**public static void** main(String[] args) { Circle circle = **new** Circle(5.0);

**double** perimeter = circle.getPerimeter();

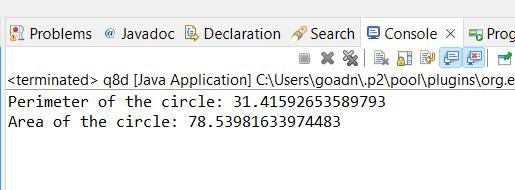
**double** area = circle.getArea();

System.***out***.println("Perimeter of the circle: " + perimeter); System.***out***.println("Area of the circle: " + area);

}

}

Output:



# Final class, abstract class, and interface

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

Ans)

**package** parctise;

**interface** Shapess {

**double** calculateArea();

**double** calculateCircumference();

}

**abstract class** TwoDimensionalShape **implements** Shapess {

**public abstract double** calculateArea();

**public abstract double** calculateCircumference();

}

**final class** Circless **extends** TwoDimensionalShape {

**private double** radius;

**public** Circless(**double** radius) {

**this**.radius = radius;

}

@Override

**public double** calculateArea() {

**return** Math.***PI*** \* radius \* radius;

}

@Override

**public double** calculateCircumference() {

**return** 2 \* Math.***PI*** \* radius;

}

}

**interface** ThreeDimensionalShape {

**double** calculateVolume();

}

**final class** Spheress **implements** ThreeDimensionalShape {

**private double** radius;

**public** Spheress(**double** radius) {

**this**.radius = radius;

}

@Override

**public double** calculateVolume() {

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

}

}

**public class** q9a {

**public static void** main(String[] args) { Circless circle = **new** Circless(5.0);

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

System.***out***.println("Circle Circumference: " + circle.calculateCircumference());

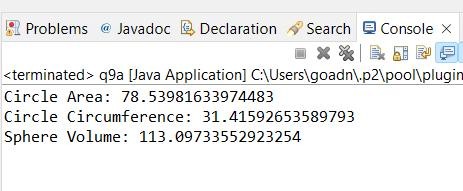
Spheress sphere = **new** Spheress(3.0);

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

}

}

Output:



* 1. 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

Ans)

**package** parctise;

**class** Circles {

**private double** radius;

**public static final double *PI*** = 3.14159;

**public** Circles(**double** radius) {

**this**.radius = radius;

}

**public double** calculateArea() {

**return *PI*** \* radius \* radius;

}

}

**public class** q9b {

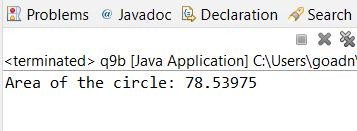
**public static void** main(String[] args) { Circles circle = **new** Circles(5.0);

System.***out***.println("Area of the circle: " + circle.calculateArea());

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**class** Animals {

**private final int** numberOfLegs; **private final int** numberOfEars; **private final int** numberOfEyes; **private final boolean** hasTail;

**public** Animals(**int** numberOfLegs, **int** numberOfEars, **int** numberOfEyes, **boolean** hasTail) {

**this**.numberOfLegs = numberOfLegs; **this**.numberOfEars = numberOfEars; **this**.numberOfEyes = numberOfEyes; **this**.hasTail = hasTail;

}

**public final void** displayLegs() { System.***out***.println("Number of legs: " + numberOfLegs);

}

**public final void** displayEars() { System.***out***.println("Number of ears: " + numberOfEars);

}

**public final void** displayEyes() { System.***out***.println("Number of eyes: " + numberOfEyes);

}

**public final void** displayTail() {

System.***out***.println("Has tail: " + (hasTail ? "Yes" : "No"));

}

}

**class** Wolf **extends** Animals {

**private final** String furColor;

**public** Wolf(**int** numberOfLegs, **int** numberOfEars, **int** numberOfEyes, **boolean** hasTail, String furColor)

{

**super**(numberOfLegs, numberOfEars, numberOfEyes, hasTail);

**this**.furColor = furColor;

}

**public void** displayFurColor() { System.***out***.println("Fur color: " + furColor);

}

}

**public class** q9c {

**public static void** main(String[] args) {

Wolf wolf = **new** Wolf(4, 2, 2, **true**, "Gray");

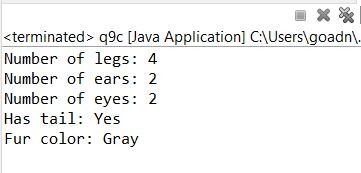
wolf.displayLegs(); wolf.displayEars(); wolf.displayEyes(); wolf.displayTail();

wolf.displayFurColor();

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**abstract class** BankAccount {

**private double** balance;

**public** BankAccount(**double** initialBalance) {

**this**.balance = initialBalance;

}

**public abstract void** deposit(**double** amount);

**public abstract void** withdraw(**double** amount);

**public double** getBalance() {

**return** balance;

}

**protected void** updateBalance(**double** amount) { balance += amount;

}

}

**class** SavingsAccount **extends** BankAccount {

**private double** interestRate;

**public** SavingsAccount(**double** initialBalance, **double** interestRate) {

**super**(initialBalance); **this**.interestRate = interestRate;

}

@Override

**public void** deposit(**double** amount) {

**double** interest = amount \* interestRate; updateBalance(amount + interest);

}

@Override

**public void** withdraw(**double** amount) { updateBalance(-amount);

}

}

**class** CurrentAccount **extends** BankAccount {

**private double** overdraftLimit;

**public** CurrentAccount(**double** initialBalance, **double** overdraftLimit) {

**super**(initialBalance); **this**.overdraftLimit = overdraftLimit;

}

@Override

**public void** deposit(**double** amount) { updateBalance(amount);

}

@Override

**public void** withdraw(**double** amount) {

**double** availableBalance = getBalance() + overdraftLimit;

**if** (amount <= availableBalance) { updateBalance(-amount);

} **else** {

System.***out***.println("Insufficient funds! Cannot withdraw.");

}

}

}

**public class** q9d {

**public static void** main(String[] args) {

SavingsAccount savingsAccount = **new** SavingsAccount(1000.0, 0.05); savingsAccount.deposit(500.0);

savingsAccount.withdraw(200.0);

System.***out***.println("Savings Account Balance: " + savingsAccount.getBalance());

CurrentAccount currentAccount = **new** CurrentAccount(2000.0, 1000.0); currentAccount.deposit(300.0);

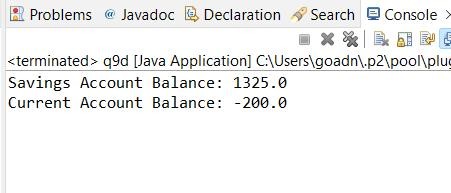
currentAccount.withdraw(2500.0);

System.***out***.println("Current Account Balance: " + currentAccount.getBalance());

}

}

Output:



* 1. 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.

Ans)

**package** parctise;

**abstract class** Shape3D {

**public abstract double** calculateVolume();

**public abstract double** calculateSurfaceArea();

}

**class** Sphere **extends** Shape3D {

**private double** radius;

**public** Sphere(**double** radius) {

**this**.radius = radius;

}

@Override

**public double** calculateVolume() {

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

}

@Override

**public double** calculateSurfaceArea() {

**return** 4.0 \* Math.***PI*** \* Math.*pow*(radius, 2);

}

}

**class** Cube **extends** Shape3D {

**private double** sideLength;

**public** Cube(**double** sideLength) {

**this**.sideLength = sideLength;

}

@Override

**public double** calculateVolume() {

**return** Math.*pow*(sideLength, 3);

}

@Override

**public double** calculateSurfaceArea() {

**return** 6.0 \* Math.*pow*(sideLength, 2);

}

}

**public class** q9e {

**public static void** main(String[] args) { Sphere sphere = **new** Sphere(3.0);

System.***out***.println("Sphere Volume: " + sphere.calculateVolume()); System.***out***.println("Sphere Surface Area: " + sphere.calculateSurfaceArea());

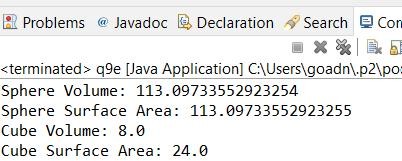
Cube cube = **new** Cube(2.0);

System.***out***.println("Cube Volume: " + cube.calculateVolume()); System.***out***.println("Cube Surface Area: " + cube.calculateSurfaceArea());

}

}

Output:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "There is nothing impossible to they who will try."

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

- All The Best -