**1)** // Step 1: Define the Person class

public class Person {

// Step 2: Declare attributes

private String name;

private int age;

// Step 3: Define the constructor to initialize the attributes

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Step 4: Getter method for name

public String getName() {

return name;

}

// Step 5: Getter method for age

public int getAge() {

return age;

}

// Main method for testing the class

public static void main(String[] args) {

// Create a new Person object

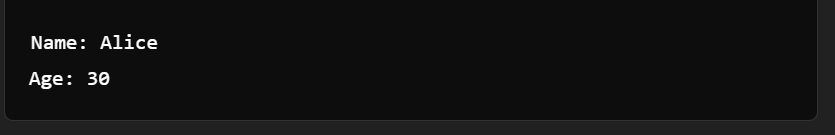
Person person1 = new Person("Alice", 30);

// Access attributes via getter methods

System.out.println("Name: " + person1.getName());

System.out.println("Age: " + person1.getAge());

}

}

**2)** public class TestMain {

public static void main(String[] args) {

// Test constructor and toString()

Employee e1 = new Employee(8, "Peter", "Tan", 2500);

System.out.println(e1); // toString();

// Test Setters and Getters

e1.setSalary(999);

System.out.println(e1); // toString();

System.out.println("id is: " + e1.getId());

System.out.println("firstname is: " + e1.getFirstName());

System.out.println("lastname is: " + e1.getLastName());

System.out.println("salary is: " + e1.getSalary());

System.out.println("name is: " + e1.getName());

System.out.println("annual salary is: " + e1.getAnnualSalary()); // Test method

// Test raiseSalary()

System.out.println(e1.raiseSalary(10));

System.out.println(e1);

}

**3)** // Step 1: Define the Circle class

public class Circle {

// Step 2: Declare the radius data member

private double radius;

// Step 3: No-argument constructor (default constructor)

public Circle() {

this.radius = 1.0; // Default radius is 1.0

}

// Step 4: Constructor with one argument (radius)

public Circle(double radius) {

this.radius = radius;

}

// Step 5: Method to calculate the circumference of the circle

public double calculateCircumference() {

return 2 \* Math.PI \* radius; // Formula for circumference: 2πr

}

// Getter method for radius (optional)

public double getRadius() {

return radius;

}

// Main method to test the Circle class

public static void main(String[] args) {

// Create a Circle object using the no-argument constructor

Circle circle1 = new Circle();

System.out.println("Circle 1 Radius: " + circle1.getRadius());

System.out.println("Circle 1 Circumference: " + circle1.calculateCircumference());

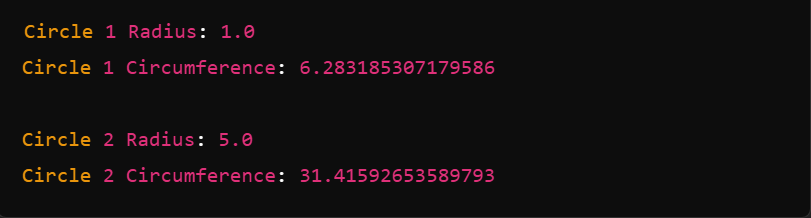
// Create a Circle object using the constructor with radius argument

Circle circle2 = new Circle(5.0);

System.out.println("\nCircle 2 Radius: " + circle2.getRadius());

System.out.println("Circle 2 Circumference: " + circle2.calculateCircumference());

}

}

**4)** // Step 1: Define the Account class

public class Account {

// Step 2: Declare data members (account details)

private String accountNumber;

private String accountHolderName;

private double balance;

// Step 3: No-argument constructor (default constructor)

public Account() {

this.accountNumber = "0000000000"; // Default account number

this.accountHolderName = "Unknown"; // Default name

this.balance = 0.0; // Default balance is 0

}

// Step 4: Constructor with two arguments (account number and holder name)

public Account(String accountNumber, String accountHolderName) {

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.balance = 0.0; // Initial balance is 0 for new accounts

}

// Step 5: Method to deposit money into the account

public void deposit(double amount) {

if (amount > 0) {

balance += amount; // Add the deposit amount to the balance

System.out.println("Deposited: $" + amount);

} else {

System.out.println("Deposit amount must be positive.");

}

}

// Step 6: Method to withdraw money from the account

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount; // Subtract the withdrawal amount from the balance

System.out.println("Withdrawn: $" + amount);

} else {

System.out.println("Insufficient funds or invalid withdrawal amount.");

}

}

// Step 7: Method to check the current balance

public double checkBalance() {

return balance; // Return the current balance

}

// Getter methods (optional, to access the account details)

public String getAccountNumber() {

return accountNumber;

}

public String getAccountHolderName() {

return accountHolderName;

}

// Main method for testing the Account class

public static void main(String[] args) {

// Create an Account object using the no-argument constructor

Account acc1 = new Account();

// Print initial account details

System.out.println("Account Number: " + acc1.getAccountNumber());

System.out.println("Account Holder: " + acc1.getAccountHolderName());

System.out.println("Initial Balance: $" + acc1.checkBalance());

// Deposit and withdraw some money

acc1.deposit(1000.0); // Deposit $1000

acc1.withdraw(500.0); // Withdraw $500

// Check the final balance

System.out.println("Final Balance: $" + acc1.checkBalance());

// Create an Account object using the constructor with parameters

Account acc2 = new Account("1234567890", "Alice");

acc2.deposit(1500.0); // Deposit $1500

acc2.withdraw(200.0); // Withdraw $200

// Print account details for second account

System.out.println("\nAccount Number: " + acc2.getAccountNumber());

System.out.println("Account Holder: " + acc2.getAccountHolderName());

System.out.println("Final Balance: $" + acc2.checkBalance());

}

}