**Question1**

package task6;

public class Person {

private String name;

private int age;

// Constructor

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Getter for name

public String getName() {

return name;

}

// Getter for age

public int getAge() {

return age;

}

}

**Main Method**

package task6;

public class Question1 {

public static void main(String[] args) {

// Creating an instance of Person

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

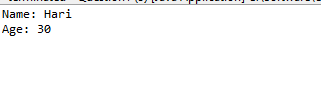
// Accessing attributes using getter methods

System.out.println("Name: " + person.getName()); // Output: Name: Alice

System.out.println("Age: " + person.getAge()); // Output: Age: 30

}

}



**Question2**

package task6;

public class Employee {

// Data members for the Employee class

private int id;

private String firstName;

private String lastName;

private double salary;

// No-argument constructor

public Employee() {

this.id = 0;

this.firstName = "Unknown";

this.lastName = "Unknown";

this.salary = 0.0;

}

// Constructor with parameters

public Employee(int id, String firstName, String lastName, double salary) {

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.salary = salary;

}

// Method to raise the salary by a specific percentage

public void raiseSalary(double percent) {

if (percent > 0) {

double increase = salary \* (percent / 100);

salary += increase;

System.out.println("Salary increased by " + percent + "%");

} else {

System.out.println("Percentage must be positive.");

}

}

// Getter methods

public int getId() {

return id;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

public double getSalary() {

return salary;

}

// Method to get the annual salary

public double getAnnualSalary() {

return salary \* 12; // Assuming monthly salary

}

}

**Main Method**

**package** task6;

**public** **class** Question2 {

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

// Create an Employee object using the parameterized constructor

Employee emp = **new** Employee(1, "Hari", "Krish", 5000);

// Display initial details

System.***out***.println("Employee ID: " + emp.getId());

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

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

System.***out***.println("Monthly Salary: " + emp.getSalary());

System.***out***.println("Annual Salary: " + emp.getAnnualSalary());

// Raise the salary by 10%

emp.raiseSalary(10);

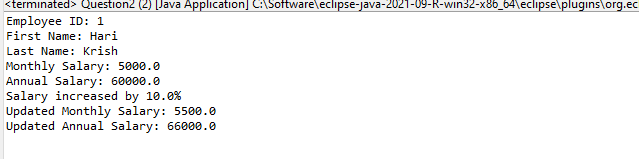
// Display updated details

System.***out***.println("Updated Monthly Salary: " + emp.getSalary());

System.***out***.println("Updated Annual Salary: " + emp.getAnnualSalary());

}

}



**Question3**

package task6;

public class Circle {

// Data member for the radius of the circle

private double radius;

// No-argument constructor that sets radius to a default value (e.g., 1.0)

public Circle() {

this.radius = 1.0;

}

// Constructor with one argument to initialize the radius

public Circle(double radius) {

this.radius = radius;

}

// Method to calculate the circumference of the circle

public double calculateCircumference() {

return 2 \* Math.PI \* radius;

}

// Getter method for radius

public double getRadius() {

return radius;

}

// Setter method for radius

public void setRadius(double radius) {

this.radius = radius;

}

}

**Main method**

package task6;

public class Question3 {

public static void main(String[] args) {

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

Circle obj1 = new Circle();

System.out.println("Default Circle Radius: " + obj1.getRadius());

System.out.println("Default Circle Circumference: " + obj1.calculateCircumference());

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

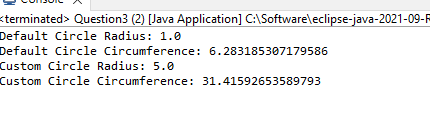
Circle obj2 = new Circle(5.0);

System.out.println("Custom Circle Radius: " + obj2.getRadius());

System.out.println("Custom Circle Circumference: " + obj2.calculateCircumference());

}

}



**Question4**

package task6;

public class Account {

// Data members for the account

private String accountNumber;

private String accountHolderName;

private double balance;

// No-argument constructor

public Account() {

this.accountNumber = "Unknown";

this.accountHolderName = "Unknown";

this.balance = 0.0;

}

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

public Account(String accountNumber, String accountHolderName) {

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.balance = 0.0;

}

// Method to deposit money into the account

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

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

} else {

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

}

}

// Method to withdraw money from the account

public void withdraw(double amount) {

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

balance -= amount;

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

} else if (amount > balance) {

System.out.println("Insufficient balance.");

} else {

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

}

}

// Method to check the current balance

public double checkBalance() {

return balance;

}

// Getter methods for account details

public String getAccountNumber() {

return accountNumber;

}

public String getAccountHolderName() {

return accountHolderName;

}

}

**Main Method**

package task6;

public class Question4 {

public static void main(String[] args) {

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

Account acc1 = new Account();

System.out.println("Account 1 - Number: " + acc1.getAccountNumber() + ", Holder: " + acc1.getAccountHolderName());

acc1.deposit(1000);

acc1.withdraw(500);

System.out.println("Account 1 Balance: " + acc1.checkBalance());

// Create an Account object using the constructor with two arguments

Account acc2 = new Account("123456789", "Hari");

System.out.println("Account 2 - Number: " + acc2.getAccountNumber() + ", Holder: " + acc2.getAccountHolderName());

acc2.deposit(1500);

acc2.withdraw(700);

System.out.println("Account 2 Balance: " + acc2.checkBalance());

}

}

