

## Java Task


1. Create a class called "Person" with attributes "name" and "age". Also create a constructor and getter methods for the attributes.

```
package project1;

public class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public static void main(String[] args) {
        Person person = new Person("Alice", 25);

        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

OUTPUT:



```
Name: Alice
Age: 25
```

2. From the below image, Create an implementation for a java class named Employee that represents an employee with attributes including ID, name, and salary. The class includes a method called raiseSalary(percent), which updates the salary by a specified percentage.

```
package project1;
```

```
public class Employee {
    private int id;
    private String firstName;
```

```
private String lastName;

private int salary;

public Employee(int id, String firstName, String lastName, int salary) {
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
    this.salary = salary;
}

public int getID() {
    return id;
}

public String getFirstName() {
    return firstName;
}

public String getLastName() {
    return lastName;
}

public String getName() {
    return firstName + " " + lastName;
}

public int getSalary() {
    return salary;
}

public void setSalary(int salary) {
    this.salary = salary;
}

public int getAnnualSalary() {
```

```

        return salary * 12;
    }

    public int raiseSalary(int percent) {
        salary += salary * percent / 100;
        return salary;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + getName() + ", salary=" + salary + "]";
    }

    public static void main(String[] args) {
        Employee emp = new Employee(1, "John", "Doe", 5000);
        System.out.println(emp);
        System.out.println("ID: " + emp.getID());
        System.out.println("First Name: " + emp.getFirstName());
        System.out.println("Name: " + emp.getName());
        System.out.println("Salary: " + emp.getSalary());
        System.out.println("Annual Salary: " + emp.getAnnualSalary());
        System.out.println("New Salary after 10% raise: " + emp.raiseSalary(10));
        System.out.println(emp);
    }
}

```

OUTPUT:

```

First Name: John
Name: John Doe
Salary: 5000
Annual Salary: 60000
New Salary after 10% raise: 5500
Employee [id=1, name=John Doe, salary=5500]

```

3. Create a class circle class with radius as data member. Create two constructors (no argument, and two arguments) and a method to calculate Circumference.

```
package project1;
```

```
public class Circle {  
    private double radius;  
    public Circle() {  
        this.radius = 1.0;  
    }  
    public Circle(double radius) {  
        this.radius = radius;  
    }  
    public double getCircumference() {  
        return 2 * Math.PI * radius;  
    }  
    public double getRadius() {  
        return radius;  
    }  
    public void setRadius(double radius) {  
        this.radius = radius;  
    }  
    public static void main(String[] args) {  
        Circle circle1 = new Circle();  
        System.out.println("Circle1 (default radius):");  
        System.out.println("Radius: " + circle1.getRadius());  
        System.out.println("Circumference: " + circle1.getCircumference());  
        Circle circle2 = new Circle(5.0);  
        System.out.println("\nCircle2 (radius = 5.0):");  
        System.out.println("Radius: " + circle2.getRadius());  
        System.out.println("Circumference: " + circle2.getCircumference());  
    }  
}
```

OUTPUT:

```
Radius: 1.0  
Circumference: 6.283185307179586  
  
Circle2 (radius = 5.0):  
Radius: 5.0  
Circumference: 31.41592653589793
```

4. Create a class Account class with all necessary parameters as data member. Create two constructors (no argument, and two arguments) and methods to withdraw and deposit balance & check balance.

```
package project1;
```

```
public class Account {  
    private String accountNumber;  
    private double balance;  
  
    public Account() {  
        this.accountNumber = "000000";  
        this.balance = 0.0;  
    }  
    public Account(String accountNumber, double initialBalance) {  
        this.accountNumber = accountNumber;  
        this.balance = initialBalance;  
    }  
    public void deposit(double amount) {  
        if (amount > 0) {  
            balance += amount;  
            System.out.println("Deposited: $" + amount);  
        } else {  
            System.out.println("Deposit amount must be positive.");  
        }  
    }  
}
```

```

public void withdraw(double amount) {
    if (amount > 0 && amount <= balance) {
        balance -= amount;
        System.out.println("Withdrew: $" + amount);
    } else if (amount > balance) {
        System.out.println("Insufficient balance.");
    } else {
        System.out.println("Withdrawal amount must be positive.");
    }
}

public double checkBalance() {
    return balance;
}

public static void main(String[] args) {
    Account account1 = new Account();
    System.out.println("Account1 (default):");
    System.out.println("Balance: $" + account1.checkBalance());
    Account account2 = new Account("123456", 1000.0);
    System.out.println("\nAccount2 (initialized with $1000):");
    System.out.println("Balance: $" + account2.checkBalance());
    account2.deposit(500);
    System.out.println("New Balance: $" + account2.checkBalance());
    account2.withdraw(200);
    System.out.println("New Balance: $" + account2.checkBalance());
    account2.withdraw(1500);
}
}

```

OUTPUT:

```

Balance: $1000.0
Deposited: $500.0
New Balance: $1500.0
Withdrew: $200.0
New Balance: $1300.0
Insufficient balance.

```