Java_Assignment-02

Instructions:

- Complete each assignment separately and test the output.
- Follow good coding practices (use meaningful variable and method names).
- Submit the completed programs before the next class.

Part 1: Methods in Java

1. Understanding Basic Methods

- Write a program that contains a method named **printMessage** that prints:

  > "Hello, welcome to Ednue Technologies!"

- Call this method in the main program to display the message.

---

2. Using Methods with Parameters

- Write a method named **addNumbers** that takes two numbers as input and prints their sum.
- Call the method with different sets of numbers.

Example:

```
addNumbers(5, 10);  // Output: Sum: 15
addNumbers(20, 30); // Output: Sum: 50
```

---

3. Methods with Return Values

- Write a method named **findSquare** that takes an integer as input and returns its square.
- Call this method in the main program and print the result.

Example:

```
findSquare(4);  // Output: Square of 4 is 16
```

---

4. Method Overloading

- Write two methods named **displayMessage**:
  - One method should print `"Hello, User!"`
  - Another method should take a name as input and print `"Hello, [name]!"`
- Call both methods and check the output.

Example:

```
displayMessage();       // Output: Hello, User!
displayMessage("Ednue Technologies"); // Output: Hello, Ednue Technolog
ies!
```

---

5. Using Methods for Calculations

- Write a program with three methods:
  - **getMaximum** → Takes two numbers and returns the larger one.
  - **isEven** → Takes a number and returns `true` if it's even, otherwise `false`.
  - **calculateFactorial** → Takes a number and returns its factorial.
- Call all three methods in the main program and print their results.

Example:

```
getMaximum(10, 20);     // Output: 20
isEven(7);              // Output: false
calculateFactorial(5);  // Output: 120
```

Part 2: Classes and Objects

6. Creating a Simple Class and Object

- Create a class named **Car** with the following details:

- Variables: `brand`, `model`, `year`
- Inside the main program, create an object of the **Car** class, assign values to the variables, and print them.

Example Output:

```
Car Brand: Toyota
Car Model: Camry
Year: 2022
```

---

7. Using Methods inside a Class

- Modify the **Car** class to include a method **displayCarDetails()** that prints the car details.
- Call this method using an object.

---

8. Class with Parameterized Method

- Create a **Rectangle** class with two variables: `length` and `width`.
- Create a method **calculateArea()** that takes **length** and **width** as input and returns the area.
- Call this method in the main program and print the area.

Example Output:

```
Rectangle Area: 50
```

---

9. Multiple Objects and Method Calls

- Create a **BankAccount** class with:
  - Variables: `accountHolder`, `balance`
  - Methods:
    - **deposit(amount)** → Adds money to the balance
    - **withdraw(amount)** → Deducts money from the balance
    - **displayBalance()** → Prints the balance

- Create **two objects** for different users and perform deposit/withdraw operations.

Example Output:

```
John's Balance: 5000
John deposited 2000. New Balance: 7000
John withdrew 1500. New Balance: 5500
```

---

10. Using Objects of Another Class

- Create a **Person** class with:
  - Variables: `name`, `age`
  - Method **displayPersonDetails()** to print the person's details.
- Create another class **Address** with:
  - Variables: `street`, `city`, `pincode`
  - Method **displayAddress()** to print address details.
- Modify the **Person** class to include an **Address** object as a variable.
- In the main program, create a **Person** object along with an **Address** object, assign values, and print the details using the methods.

Example Output:

```
Person Name: Arun, Age: 25
Address: 123, MG Road, Chennai - 600001
```

- If you have doubts, feel free to ask!

Happy coding! 🚀🔥
