Object Oriented Programming
Lab 02: Classes

January 12, 2023

1 Lab Goal

In this lab we will work with classes in Java. We will first explain how a class is created and then add a few variables and functions to get a good grip on the concept. We will then do a few problems to test our understanding of the topic.

You can access Java Documentation at this link.

2 Example

First, we define the class using the class keyword, followed by the name of the class (in this case, "Example"). The class definition should start with an uppercase letter and should be in camelCase.

```
class Example {
    // class body goes here
}
```

Next, we can add class variables (also known as fields) and class methods inside the curly braces of the class definition. For example:

```
class Example {
    // class variables
    int x;
    String name;

    // class methods
    void setX(int x) {
        this.x = x;
    }

    void setName(String name) {
        this.name = name;
    }
}
```

To create an object of the Example class, we use the new keyword and call the class constructor. For example:

```
Example ex = new Example();
```

Once we have an object, we can use the class methods to set the values of the class variables. For example:

```
ex.setX(5);
ex.setName("John");
```

Finally, we can access the class variables using the object. For example:

```
System.out.println(ex.x); // prints 5
System.out.println(ex.name); // prints "John"
```

The important points that you should know are:

- The default constructor is provided by Java Compiler if you do not define any constructor in your class.
- The constructor is a special method that is used to initialize the state of an object when it is created. It is typically used to set the initial values of class variables, and it has the same name as the class.
- Also, you can have multiple constructors with different argument to initialize the object in different ways.

3 Problems

3.1 Problem 1 - Creating a Book

Create a class called "Book" that has the following properties:

- Title (string)
- Author (string)
- ISBN (string)
- Number of pages (int)

Create a constructor that initializes the properties of the book, and create a method called "summary" that returns a string containing the title, author, ISBN, and number of pages of the book.

Create a main method that creates an object of the Book class and sets its properties using the constructor. Then, call the summary method and print the result.

Here is an example of one of the outputs:

```
Title: Harry Potter and the Sorcerer's Stone
Author: J.K. Rowling
ISBN: 1234567890
Number of pages: 309
```

3.2 Problem 2 - Creating an Array of Objects

Create a class called "Car" that has the following properties:

- Make (string)
- Model (string)
- Year (int)
- Color (string)
- Number of wheels (int)
- Number of doors (int)

Create a main method that creates an array of 3 Car objects and sets their properties using the constructor. Then, call the details method for each object and print the result.

Example output might look like:

```
Car 1:
Make: Ford
Model: Mustang
Year: 2020
Color: Red
Number of wheels: 4
Number of doors: 2

Car 2:
Make: Honda
Model: Civic
Year: 2019
Color: White
```

```
Number of wheels: 4
Number of doors: 4

Car 3:
Make: Tesla
Model: Model S
Year: 2018
Color: Black
Number of wheels: 4
Number of doors: 4
```

3.3 Problem 3 - Passing Objects to Objects

Create two classes, "Person" and "Pet", where a "Person" class has the following properties:

- Make Name (string)
- Make Age (int)
- Make Pet (Pet)

And a "Pet" class has the following properties:

- Make Name (string)
- Make Breed (string)
- Make Age (int)

Create a constructor for each class that initializes the properties.

Create a method in each class called "getDetails" that returns a string containing the person and their pet details.

Create a main method that creates an object of the "Pet" class and sets its properties using the constructor. Then, create an object of the "Person" class and sets its properties using the constructor, including the "Pet" object as an argument. Finally, call the "getDetails" method and print the result.

Here's an example of what the output might look like:

```
Name: John
Age: 25
Pet Name: Fido
Pet Breed: Labrador
Pet Age: 5
```