

## Objectives:

The students should learn how to:

1. Describe objects and classes, and use classes to model objects.
2. Use UML graphical notation to describe classes and objects.
3. Demonstrate how to define classes and create objects.
4. Create objects using default constructors.
5. Access objects via object reference variables.
6. Define a reference variable using a reference type.
7. Access an object's data and methods using the object member access operator (.).

## Submission rules:

- **Due date: Saturday, October 30th, 2021 at 11:59 pm**
- You can discuss answers with your colleagues. **But no copying.**
- Submit it to lms.ksu.edu.sa. **Email submissions will not be accepted.**
- All classes in one java project. The project name must be: **Lab06 ID FirstName LastName.zip**. For example:  
Lab06\_123456789\_Marwan\_Almaymoni.zip
- Use the default package.
- Write your name and university ID as a comment at the start of all java files.

## Lab Exercise 1

### Part1

Write a Java program that prints the name and age of two students.  
Name your class **TestStudent**.

Sample Run

```
Student Name: Saleh
Student age: 22
-----
Student Name: Ali
Student age: 25
```

### Part 2

In the previous program, you have defined two variables to store the names of two students and another two to store the ages. But if we want to have more than two students, then we'll have variables like `student1Name`, `student2Name`, `student3Name`... etc. In this case, it is better to use objects.

Rewrite the previous program using classes and objects. The new program will contain one class named **Student** and two objects of this class. Write the program by completing the following pseudo-code:

Class **Student** pseudo-code:

```
class Student {
    // data members
    public String name;
    //define instance variable age as integer
    /* modifier datatype variable name */
}
```

Class **TestStudent** new pseudo-code:

```
public class TestStudent {
    public static void main(String[] args) {
```

```

        // creating objects of student

        //create first student object named s1
        Student s1 = new Student();

        /* create second student object named s2 */

        //assign first student name "Saleh"
        s1.name = "Saleh";
        //assign first student age 22
        s1.age = 22;

        /* assign second student name "Ali" */
        /* assign second student age 25 */

        //printing students information

        //print first student name
        System.out.println("Student name:\t" + s1.name);
        /* print first student age */

        /* print second student name */
        /* print second student age */

    }
}

```

### Sample Run

```

Student Name:    Saleh
Student age:     22
-----
Student Name:    Ali
Student age:     25

```

## Part 3

Now write the same program and add a new instance variable of type double to store the student's GPA.

Class **Student** new pseudo-code:

```

class Student {
    // data members
    // define instance variables name, age, GPA
    /* modifier datatype variable name */
    /* modifier datatype variable name */
    /* modifier datatype variable name */
}

```

Class **TestStudent** new pseudo-code:

```
public class TestStudent {  
    public static void main(String[] args) {  
        //creating objects  
        /*create object s1*/  
        /*create object s2*/  
  
        //assign values to the instance variables of the two objects  
        /* */  
        //print the two students information  
        /* */  
    }  
}
```

Sample Run

```
Student Name: Saleh  
Student age: 22  
Student GPA: 4.25  
-----  
Student Name: Ali  
Student age: 25  
Student GPA: 4.5
```

## Part 4

Now we would like to change the previous program to allow the user of the program to enter a course name and then we assign all students to this course. Start by adding a property called course of type String to class Student. Add a String variable named course. The value of “course” is read from the user at the beginning of the program. After that, assign this course to each student that you create.

Class **Student** new pseudo-code:

```
class Student {  
    // data members  
    // define instance variables name, age, GPA, course  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
}
```

Class **TestStudent** new pseudo-code:

```
public class TestStudent {  
    public static void main(String[] args) {  
        /*define variable course and read its value*/  
  
        //creating objects  
        /*create object s1*/  
        /*create object s2*/  
  
        //assign values to the instance variables of the two objects  
        /* */  
        //print the two students information  
        /* */  
    }  
}
```

Sample Run

```
Enter course name: CSC111 ↵  
Student Name: Saleh  
Student age: 22  
Student GPA: 4.25  
Student course: CSC111  
-----  
Student Name: Ali  
Student age: 25  
Student GPA: 4.5  
Student course: CSC111
```

## Part 5

Now write the same program and add a method `printInfo()` to print student's information. The method does not receive any parameter and does not return any value.

Class **Student** new pseudo-code:

```
class Student {  
    // data members  
    // define instance variables name, age, GPA, course  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
    /* modifier datatype variable name */  
}
```

```

//Define method printInfo that is used to print Student's information
/* modifier returntype*/ printInfo(/*parameters*/)
{
    /* body */
}
}

```

Class **TestStudent** new pseudo-code:

```

public class TestStudent {
    public static void main(String[] args) {
        //creating objects
        /*create object s1*/
        /*create object s2*/

        //assign values to the instance variables of the two objects
        /* */
        //print the two students information
        /*call printInfo on each of the two objects*/
    }
}

```

Sample Run

```

Enter course name: CSC111 ↵
Student Name: Saleh
Student age: 22
Student GPA: 4.25
Student course: CSC111
-----
Student Name: Ali
Student age: 25
Student GPA: 4.5
Student course: CSC111

```

**Done...**