**DPIT111 Assignment 1**

**Week 2-3**

**Objectives**

# Get familiar with the fundamental OOD concepts

* Get familiar with UML class and object diagrams
* Get familiar with different data types, variables, operators and arithmetic expressions

# Get familiar with method, parameter list and return type

* Distinguish local variables from parameters
* Get familiar with this reference variable and program with more than one class

**Assignment 1 (12%)** is marked on a scale from 0-12 marks.

**Important notices:**

This assignment is scheduled to be done in Week 2-3 (Monday 24 October 00:00 - Sunday 6 November 23:55). The questions are designed based on the knowledge introduced in weeks 1-3. Comments and feedbacks for your work can be obtained from your assessor. You may be asked questions about your code. Your marks will be deducted if you could not answer the questions presented by the assessor.

1. Students must submit all assessment tasks by the due date or penalties will be applied.

|  |  |
| --- | --- |
| **No. of calendar days late** | **Penalty** |
| 1 | 10% of the original mark. |
| 2 | 20% of the original mark. |
| 3 | 30% of the original mark. |
| More than 3 | A mark of zero will be awarded for the assessment. |

1. Make sure that you include following personal information in your file for questions 6 and 8. A warning will be given for assignment 1 if this is missing.

/\*------------------------------------------------------

My name:

My subject code: DPIT111

My student number:

My email address:

Assignment number: 1

Question number:

-------------------------------------------------------\*/

1. Ensure both question 6 and 8 in your submission are runnable, **compiling error in your file will result in 0 points for that question**. If it is runnable, though it fails to realize all functionalities, you can get partial marks for what you have done.

**PART 1: UML Diagrams**

**Question 1 (1 mark):** Assume that you need to implement a ***class*** called Sport. Its incomplete UML diagram is shown below **on the right**:

|  |
| --- |
| class **Sport** { private String **name**; private int **numTeams**; private double **registrationCost**; public static void **main** (String[] args) { // code not included } public static void **printCost** (double cost) { // code not included }} |

|  |
| --- |
|  |
| name:numTeams:registrationCost: |
| main (String[]): voidprintCost |

**Complete the UML diagram (on the right)**. Type your answer directly in the box provided. Use the following questions as a guide and think about how you represent each of these things in your UML diagram:

*- What is missing? (class name, attributes or methods)*

*- Which attributes or methods are* ***public*** *or* ***private****?*

*- Which attributes or methods are* ***static****?*

**Question 2 (1 mark) :** Provide an example of an ***object*** sport1 that is an instance of the above class. Specify appropriate values for each attribute (e.g. name = …). Draw the corresponding UML diagram in a simplified object view (Object name and attributes only. No methods are needed for a simplified UML object) in the red box provided below.

|  |
| --- |
|  |
|  |

**Question 3 (1 mark) :** Assume that your program has the following classes:

 House, Room, Kitchen, Bathroom.

Which classes are related and what is the relationship type between them? (association/composition/inheritance). Draw a UML diagram to show these relationships. You only need to show the classes (not the attibutes or methods) and the correct relationship types.

<save your UML diagram as a jpg, png or something similar and paste it here>

**Question 4 (2 mark) :** The following class has errors.

public class JavaApplication1 {

 public static void main(String[] args)

 {

 int num1 = 55;

 int num2 = num1 + 3;

 System.out.println("num2 = ", num2);

 }

}

1. Explain what is wrong (don’t just repeat what the compiler says).
2. Paste below a correct version of the program with the error fixed.

**Question 5 (1 mark) :** The following class calculates the value “num2”.

public class JavaApplication1 {

 private int num;

 public static void main(String[] args) {

 int num1 = 5;

 **int num2 = num1 + 2 + num1++;**

 System.out.println("num1 = " + num1 + ", num2 = " + num2);

 }
}

The program gives an output of “num1 = 6, num2 = 12”. Explain the calculation order of the expression (shown in bold).

**Question 6 (2 mark) :** Write a program to read in two inputs x and y. Use these values to calculate z using the formula shown below.

***REQUIREMENTS***

* The user input is always correct (**input verification is not required**).
* Assume each value is in **double** type.
* Your code must use the **Scanner** class to read the user input.
* **The calculation must be done in a separate method** called calculateValue(double x, double y) which returns the value of z.
* You **must** use Math.pow()in your calculation for both x2 and y5.
* Your code must work exactly like the following example (**the text in bold indicates the user input**).

$$z=\frac{0.2+ x^{2}+y}{3x +y^{5}}+4.2$$

***Example of the program output:***

Enter value x: **2.1**

Enter value y: **3.6**

The value of z is 4.213437829562361

**Question 7 (1 mark) :** Draw the UML class diagram of your solution to **Question 6** in the space below.

|  |
| --- |
|  |
|  |
|  |

**Question 8 (3 mark) :** Define ***two classes***. The first class contains the main method which then calls a method from the second class. The second class contains a method calcSquare that takes 1 integer argument and returns the square of that number.

|  |
| --- |
| Question8 |
|  |
| + main(String[]): void |

|  |
| --- |
| Square |
|  |
| + calcSquare(int): int |

***REQUIREMENTS***

* The user input is always correct (**input verification is not required**).
* Your code must use the **Scanner** class to read the user input (an integer).
* Your code **must** include 2 classes.
* Your code must work exactly like the following example (**the text in bold indicates the user input**).

***Example of the program output:***

Enter an integer: **5**

The square of 5 is 25

***Submission requirements:***

1. Put the solutions to Question 1-5, and 7 in a word file (.doc or .docx), put both Question 6 and 8 in a separate text file (.txt, .rtf, .rtx or .java), and upload all of these 3 files to the assignment 1 submission link. The submission link is called **Assignment 1 - Submission** under the **Assessments** section on the subject Moodle site.
2. Submission via email is not acceptable without permission.
3. Enquiries about the marks can only be made within a maximum of 1 week after the assignment results are published. After 1 week the marks cannot be changed.

**Mark deductions:** late submission, compilation errors, incorrect result, program is not up to spec, poor comments,poor indentation, meaningless identifiers, required numeric constants are not defined, the program uses approaches which has not been covered in the lectures. The deductions here are merely a guide. Marks may also be deducted for other mistakes and poor practices.

**END OF THE ASSIGNMENT**