|  |  |
| --- | --- |
| **Programming II**  Diploma in IT / DS / CSF  Year 1 (2023/24) Semester 2 | Week 3 |
| **2** hours |
| **Practical 3: Introduction to OOP** | |

**OBJECTIVES**

At the end of this exercise, you should be able to implement C# programs to

* Interpret a class element diagram and complete a C# class
* Create objects from a class

|  |
| --- |
| **IMPORTANT**   * Create a folder, **week03.** * Create a new Console App (.NET Core) project, **Snnnnnnnn\_StudentApp**, in the **week03** folder created above *(note:* ***Snnnnnnnn*** *is your Student Number)*. * At the end of the session, copy the folder **week03** folder (which contains all your work) to PRG2 Brightspace. |

1. Download the file "Student.cs" from PoliteMall.
2. Add a class in the Console App project that you created.

Graphical user interface, application

Description automatically generated

1. Add a class "Student.cs"

Graphical user interface, text, application

Description automatically generated

1. Remove the code “"internal class Student".

Graphical user interface

Description automatically generated

1. Paste the code from step 1 above into the Student class

Graphical user interface, text, application

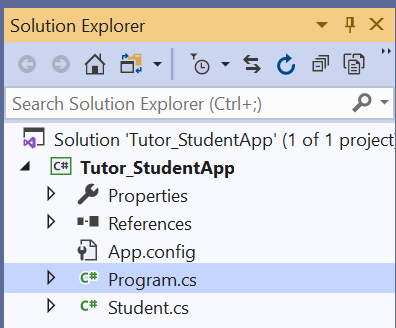
Description automatically generated

.

1. Study the class element diagram of Student class below, complete the missing attributes and Properties as well as the incomplete student class constructor.

|  |
| --- |
| Student |
| -id:int  -name:string  -tel:string  -dateOfBirth:DateTime |
| +Student(i:int, n:string, t:string, dob:DateTime) |

1. Open the Program class from the Solution Explorer and complete the following tasks.



* 1. Create 5 Student objects with your own choice of data (the first one is shown in the screenshot below).

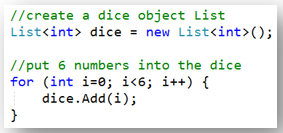
Text

Description automatically generated

* 1. Display the information for each student using Console.WriteLine(). Screenshot below shows a sample of the output:

|  |
| --- |
| ID Name Tel Date of Birth  1 John Tan 88552211 13/10/2000  2 Peter Lim 85678141 01/11/2001  3 David Chan 88555461 03/01/2000  4 Muhammed Faizal 98762211 07/05/2000  5 Esther Eng 83352245 09/08/2000 |

* 1. The example below shows how to use a C# collection List that contains integer values.



Likewise, C# collection **List** can be used to store objects.

Create a C# collection **List** called studentList that contains Student objects. Add the 5 Student objects created in part (a) above into the studentList.

* 1. Write a method with the following method signature to display the contents of the collection sList by using a loop (format of the output same as part (b) above):

void DisplayOutput(List<Student> sList)

Call the method in the Main program to display the contents of studentList.

* 1. Write a method with the following method signature to prompt the user for the student ID, name, phone and date of birth for one student, create the Student object and return the Student object.

Student GetStudent()

Call the method in the Main program and add the student to the list studentList created in part (c) above.

Call the DisplayOutput() method in the Main program again to display the contents of studentList.

* 1. The data file Students.csv contains a list of student information. Download the file from PoliteMall. Examine the file using Notepad.

Text, application

Description automatically generated

* Add the data file to your project in Visual Studio.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Set the property of the file to “copy always” or “copy if newer”. This allows the file to be copied from the project folder into the bin/Debug/.. folder at runtime.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Add codes to your program to read all the data from the file. Create the Student objects and store them in a List called studentList2. Call the method DisplayOutput() defined in part (d) above to display the contents of studentList2.

**SalesEmployeeApp (Advanced)**

1. Create a new Console App (.NET Core) project, **Snnnnnnnn\_SalesEmployeeApp**, in the **week03** folder created above *(note:* ***Snnnnnnnn*** *is your Student Number)*.
2. Write the code to implement the SalesEmployee class based on the class diagram given below:

|  |
| --- |
| SalesEmployee |
| –id:int  –name:string  –basicSalary:double  –sales:double |
| +SalesEmployee(i:int, n:string, bs:double, s:double) |

4. Write the code in the main program (program.cs) to do the following:

1. Create 5 SalesEmployee objects with with the following information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 101  Angie  1200  15000 | 105  Cindy  1000  12000 | 108  David  1500  20000 | 112  Jason  3000  30000 | 127  Vivian  2000  25000 |

1. Create a dictionary, employeeDict, and add the 5 SalesEmployee objects to the dictionary. Each object is to be stored with its id as key.
2. Display the contents of the employeeDict.

**Plagiarism Warning:**

**If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical. Disciplinary action may also be taken.**

**Similar action will be taken for student who allows other student(s) to copy his/her work.**