**CSC 7050 Programming 1 - OOP**

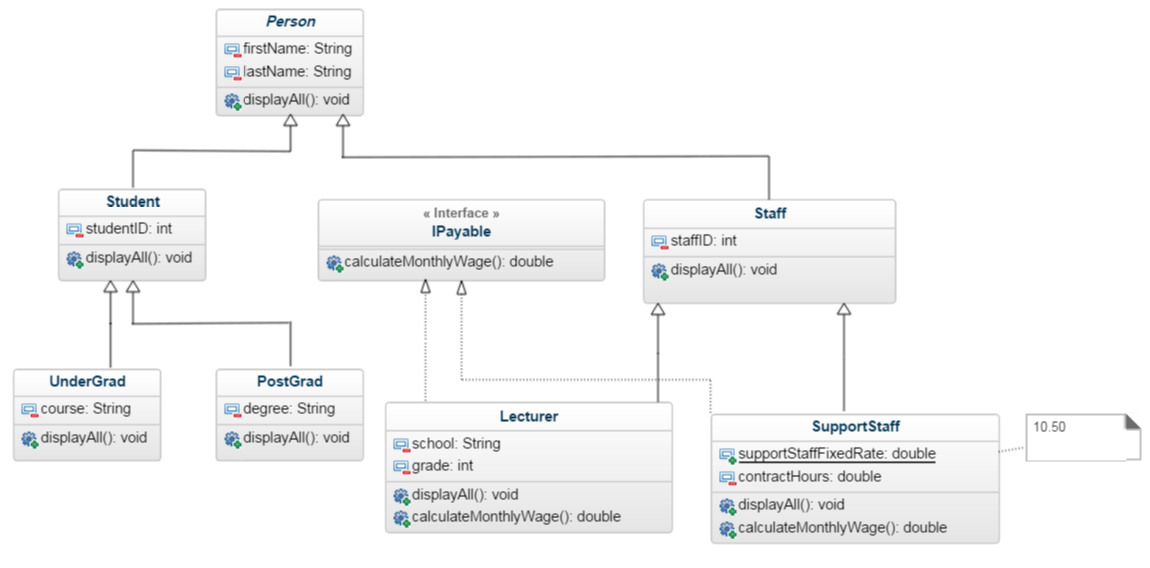
Time limited 2 hours

**Task**

You may complete this question using an appropriate IDE. When completed you should compress (zip) the finished IDE project (including source code) and submit via QOL Assignment tool and email to [**aidan.mcgowan@qub.ac.uk**](mailto:aidan.mcgowan@qub.ac.uk)**.** The project solution and package folder should both be named **p1p3<your student id>** e.g. p1p35048201.

**Part One – Creation of classes**

From the following UML class diagram create the code for a University Personal application. Ensure that you use appropriate Object Oriented principles, including Inheritance, Encapsulation, Abstraction and Polymorphism.

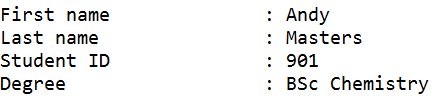


[Continued overleaf]

**Part Two - Implementation details**

**Add the following implementation details to the classes.**

1. All **displayAll** methods should output to screen the instance variables of the particular class. For example for a **PostGrad** instance it should look like this :



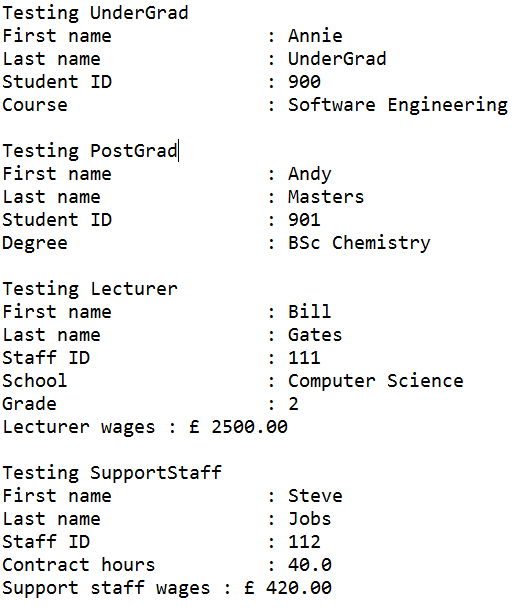
1. **SupportStaff class**
   1. **contractHours** : allowable values 1 to 160 inclusive (default is 0)
   2. **calculateMonthlyWage**   
      return value is dependent on *contractHours* \* *supportStaffFixedRate*
   3. ***supportStaffFixedRate*** *is* **10.50**
2. **Lecturer class**
   1. **grade** : allowable values 1,2,or 3 (default is 1)
   2. **calculateMonthlyWage** – return value is dependent on grade

|  |  |
| --- | --- |
| Grade | Monthly Salary |
| 1 | 2000 |
| 2 | 2500 |
| 3 | 3000 |

**Part Three**

Create an **Admin** class (with a main method) and conduct a test of the application.

An exact output formatting shown for the test run is not important but the test data required is:



You manager has suggested other test cases may be useful especially of the error conditions e.g. setting a grade of the Lecturer to outside the accepted range.

[end]