Instruction to candidates:

Your program code and output for each of Task 1 to N should be saved in a single `.ipynb` file using Jupyter Notebook. For example, your program code and output for Task 1 should be saved as:

`Task1_<your name>_<centre number>_<index number>.ipynb`

Make sure that each of your `.ipynb` files shows the required output in Jupyter Notebook.

1 Name your Jupyter Notebook as

`Task4_<your name>_<centre number>_<index number>.ipynb`

The examinations department of a school needs to keep long-term records of the overall examination achievements of its students.

Students at the school have two main choices.

1) They can choose to take a variety of subjects and achieve an Academic Diploma, which gives them the opportunity to go to university.

2) Alternatively, they can achieve a Skills Certificate where they focus on one particular area (such as IT). This gives them the necessary skills to start a career in their chosen area.

The examinations department decides to store the following data:

- `StudID` – a six digit identification number used to uniquely identify a particular student. The first four digits represent the year that the student started at the school and the last two digits are used to make the `StudID` unique e.g. 201804.

- `Name` – name of the student and is at most 30 characters. Only the first 30 characters are used for students whose name is longer than 30 characters

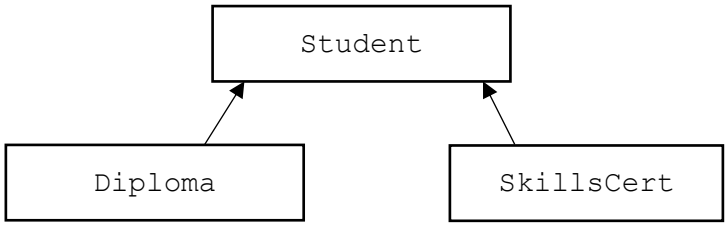
- `StudType` – type of student and can have the values `'D'` or `'S'`, representing students on the Academic Diploma track or the Skills Certificate track respectively.

- `SkillArea` – a text string indicating the area which a student on the Skills Certificate track acquired skills in. It can have one of three values: `'IT'`, `'Business'` or `'Accountancy'`.

- `NoOfSub` – number of subjects studied by a student on the Academic Diploma track.

- `Result` – a single character used to indicate the overall grade awarded.
  - For students on the Skills Certificate track, the grades could be Distinction (`D`), Merit (`M`), Pass (`P`) or Fail (`F`).
  - For students on the Academic Diploma track, the grades could be one of the letters `A` to `F`. Grades `A` to `E` are pass grades. Grade `F` is a fail grade.

The program design for a solution to this problem is to be implemented using object-oriented programming with the following three classes:



For each of the sub-tasks, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
In [1] :    # Task 4.1
            Program code

            Output:
```

Task 4.1

Write program code to implement the classes `Student`, `Diploma` and `SkillsCert`.

[6]

Task 4.2

The file `STUDENT.txt` contains details of each student. The format of each student record in the file is as follows:

`<StudID>|<Name>|<StudType>|<SkillArea>|<NoOfSub>|<Result>`

- `SkillArea` would have the value `'Diploma'` for students on the Academic Diploma track.
- `NoOfSub` would have the value `0` for students on the Skills Certificate track.
- `Result` is left blank initially.

By making use of the classes implemented in **Task 4.1**, write a module `ENTER_RESULT`, which may be a function or a procedure. When called, the module will ask the user for a particular `StudID` whose result is to be entered. Using the `StuID` provided by the user, the corresponding student record will be located in `STUDENT.txt`. The student data will then be displayed to the user. The user is then allowed to enter the `Result` for the student. The amended record will be stored back in `STUDENT.txt`.

The module should include mechanisms to validate the `StuID` input provided by the user, as well as the `Result` entered.

If the `StuID` provided by the user does not exist, an appropriate message should be displayed.

Test your module by running it **three** times. Use the following data input:

StudID	Result
201701	A
201801	B
201901	M

[10]

Task 4.3

A report should be generated and displayed, which will list the students whose result has still not been entered into the `STUDENT.txt` file. The report will list, for each different starting year:

- `StudID`
- `Name`
- `StudType`
- `SkillArea` or `NoOfSub` depending upon the value of `StudType`

In addition the number of each student type for each year will also be output.

A sample output is shown below.

```
Year: 2017
------------------------------------------------------------
201715          FLoo            D               6
201708          Blang           D               5
201710          LArms           S               IT
Diplomas: 2
Skills: 1

Year: 2018
------------------------------------------------------------
201813          EJean           D               7
201817          ABright         D               7
Diplomas: 2
Skills: 0

Year: 2019
------------------------------------------------------------
201905          Alfie           S               Business
201903          GKoh            D               8
Diplomas: 1
Skills: 1
```

Write program code to generate the report.

[10]

Save your Jupyter Notebook for Task 4.