I_Do 1

The purpose of this programing activity is to enable you to evaluate whether you acquired the OOP skills required for the rest of the module, focusing on ***class design, inheritance and polymorphism.***

In your career as programmer, you will encounter many topics to develop programs for, which you have no knowledge of. Perhaps you will have to write a program on Anaesthetics used in Hospitals or Leaf Growth in Plants…. A good programmer is always a quick learner – it is why we want to be programmers – to learn from all these fields. So in this assignment you may encounter something you never knew….

Your program must assist a corporate organisation to keep track of the *annual depreciation* of their computers.

What is *depreciation*? It is the loss of value of assets – such as cars and computers – computed annually (per year).

This company groups their computers into two groups: **Laptop Computers** and **Desktop Computers**. They use different formulas to compute the depreciation of laptops and desktops.

- They use a standard depreciation formula for their ***desktop*** computers. The formula is:

  Annual depreciation amount = (Purchasing Cost – Residual Value) / Useful life

  - The *residual value* is entered when the computer is bought. It is the estimated value that the computer can be sold for, to recover some cost when the computer has no longer any value for the organisation.
  - The *useful life*, also entered when the computer is bought is the expected lifespan of the computer.

The following table contains example data for you to be able to test your program.

Stock Code of Computer	Purchasing Cost	Residual Value	Useful life	Computed value of annual depreciation
AC201245	15000	1000	7	2000
DD120342	25000	7000	6	3000

Final notes about desktops: The company does **not** store the information of peripheral devices such as screens / keyboards etc. along with the desktop. They view them as separate assets, which you may ignore. They do not own any "All-in-one" computers. They also store the *office number* where the computer is installed, in order to be able to inspect the asset.

- From their experience, the company learned that the value of ***laptops*** decreases at percentage of the purchasing cost and that it is more difficult to estimate the residual value for laptops in general. They also learned that large screen laptops (screen size larger or equal to 12 inches) lasts longer than small screen laptops (smaller than 12 inches).

They developed their own formula:

For larger laptop they compute their annual depreciation at 20% of the purchasing cost and the smaller ones at 25%.

The following table contains example data for you to be able to test your program.

Stock Code of Computer	Purchasing Cost	Screen Size	Computed value of annual depreciation
ZC202247	15000	14 inch	3000
FF193341	10000	11 inch	2500

Final notes about laptops: Laptops are **not** allocated to offices. The company is very aware of the theft risk of their laptops, so they have all-risk insurance policy for every laptop. You need to store the insurance premium which is a fixed amount for each laptop. The amount is entered when the laptop is bought.

All computers are identified by a *Stock Code* containing 8 Characters. All computers have a responsible person assigned to it. You only need to store the 8-character *staff number* of this person. The company also stores information about the computer such as *serial number*; *CPU type*; *CPU speed* and *RAM amount*.

CMPG221: I_DO 1 task:

You need to develop the super and subclass structure for this company.

Your ***super class*** should contain an abstract method called: **AnnualDepreciation()** that computes the depreciation according to the type of computer. Carefully think which attributes should be in which class. You need to specify a **toString()** method for each of the subclasses.

Similar to the class example, you do **NOT** need to develop a user interface – you can enter the data in the test program. Your ***test program*** should:

1. Create objects with the data from the tables above. You can select your own values for the fields that are general to all computers.
2. Demonstrate Static binding – Please add clear comments to your code and output to show that you do static binding. You may use other dummy data for this purpose.
3. Create an array of all the computers and compute the annual depreciation for each computer. You have to use one array for all computers. Therefore, you need to use polymorphism. Plan your output to provide a neat list of all your data.

Remember this is not a group assignment.
Enjoy!!!