**Recommended programming language is C++**

Design an application that has the following features:

1. An interface that contains at least 3 prototypes / method signatures

2. A parent class (A) which:

a) Shall implement the interface defined in point 1

b) Defines at least 3 attributes, of which one attribute is of the class type defined in point 3)

c) Overloads 3 constructors for the 3 attributes

d) Has at least one method apart from set and get for attributes that will be overridden in classes B and C from point 4.

3. A class to be used in composition in the basic class (A)

4. Two classes (B and C) that will inherit the base class (A). These classes will in turn define at least 2 other private attributes in addition to the attributes inherited from the base class, as well as a protected attribute.

5. A class (D) inheriting class B, and which will define 3 other attributes (1 private and 3 public)

6. Define a class (E) through which to highlight the friendship defined at the level of class (B-E) and at the level of methods (C-E)

Tips to follow:

• Use set and get methods where appropriate

• Use constructor delegation and initialization lists

• Highlight the dynamic polymorphism

• Separate definition (.h) and implementation (.cpp) into separate files

• The class hierarchy has to make sense