IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2017

EIE PART II: MEng, BEng and ACGI

Corrected Copy

SOFTWARE ENGINEERING 2: OBJECT-ORIENTED SOFTWARE ENGINEERING

Wednesday, 7 June 10:00 am

Time allowed: 2:00 hours

There are THREE questions on this paper.

Answer ALL questions.
Q1 carries 40% of the marks. Questions 2 and 3 carry equal marks (30% each).

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : M. Cattafi
Second Marker(s) : K. Fobelets

SOFTWARE ENGINEERING 2: OBJECT ORIENTED SOFTWARE ENGINEERING

1. This is a general question about C++ and Object Oriented Software Engineering.

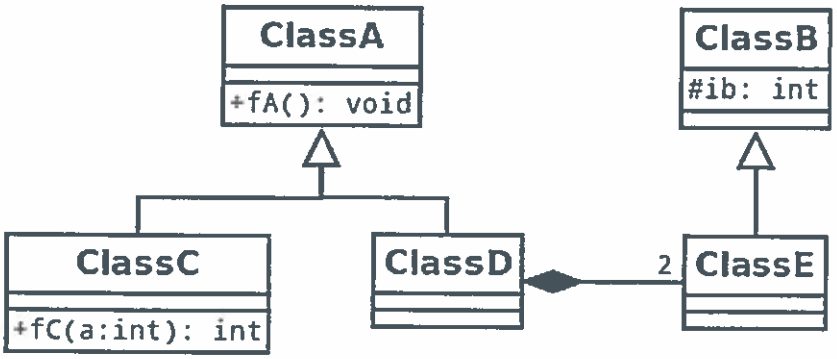

Figure 1.1 An UML diagram

a) Describe in words the software architecture represented in the diagram in Fig. 1.1.

[6]

b) Write C++ declarations for all the classes in the UML diagram in Fig. 1.1. The declarations can be kept to the essential skeleton (e.g. constructors can be omitted) but all the elements related to available information (including relationships) should be included.

[9]

c) We want to create an example of polymorphism based on `ClassB` and `ClassE` from the software architecture in Fig. 1.1.

i) Explain why this requires a change in these classes.

[3]

ii) Write C++ code for an amended version of the declaration of the classes that would be suitable for an example of polymorphism.

[3]

iii) Write C++ code that could e.g. be in the `main` and that would represent an example of polymorphism based on `ClassB` and `ClassE`.

[3]

d) i) List all the access modifiers available in C++ for member data and member functions and explain their meaning.

[4]

ii) Explain which access modifiers are most often associated respectively to member data and member functions and why.

[4]

e) Consider the insertion (<<) operator.

i) Describe the cases (if any) in which its overloading is defined as member function and the cases (if any) in which it is defined as global function.

[3]

ii) Discuss why it is so.

[5]

2. This question deals with C++ templates and the Standard Template Library.

a) i) Write a template function which, given in input an `std::list` containing elements of a generic type, changes the list so that its minimum and maximum elements are swapped.

[12]

ii) Explain which are the characteristics that the generic type mentioned above (i.e. the type of the elements contained in an `std::list` given in input to this function) must have.

[6]

b) i) Discuss the main difference between the iterators that can be used (respectively) with `std::vector` and with `std::list`.

[2]

ii) Illustrate your answer using C++ code.

[5]

c) Explain what is a `const_iterator` and when it is used.

[5]

3. This question deals with Java.

a) Compare and contrast how the process of turning code into an executable program works in C++ and in Java. Explain how this affects the portability of Java applications.

[8]

b) Write Java code (roughly equivalent to the C++ code requested in question 1.b) for all the classes in the UML diagram in Fig. 1.1. The body of the functions can be kept empty.

[8]

c) Explain why Java doesn't have the initialisation list feature that C++ has.

[7]

d) Explain what `interfaces` are in Java and how they are used.

[7]