

Corrected copy

SOFTWARE ENGINEERING 2: OBJECT ORIENTED SOFTWARE ENGINEERING

1. This is a general question about C++ and Object Oriented Software Engineering.
- a) Consider an application domain related to wheeled vehicles, in particular dealing with bicycles and cars. Bicycles have a front and a back wheel. Cars have two front wheels and two back wheels. For all wheeled vehicles we want to be able to know the cost of their set of wheels (as the sum of the cost of each of their wheels).
 - i) Describe in words how you would model this domain in an object oriented architecture.

[8]
 - ii) Draw a UML class diagram of the architecture.

[8]
 - iii) Write a declaration for all the classes. The declarations can be kept to the essential skeleton (e.g. constructors can be omitted) but all the relevant elements needed in order to express the architecture should be included. Include appropriate implementation(s) for the operation related to the cost of the set of wheels of wheeled vehicles.

[10]
 - b) Explain the concept of function binding, distinguishing between "early" and "late" binding. Discuss how this concept relates to polymorphism in C++, illustrating your answer with an example in C++ code.

[8]
 - c) Explain what are #include guards in C++ header files and why they are important. Illustrate your answer with an example.

[6]
2. This question deals with C++ templates and the Standard Template Library.
- a)
 - i) Write a C++ template function that swaps the contents of two objects of the same generic type.

[4]
 - ii) Using the swap function defined above, define another function that takes in input a `std::vector` containing elements of a generic type and changes it so that after the function call its elements are in reversed order. The implementation must make use of iterators.

[9]

- b) Explain why the usual separation of function declarations from their corresponding definitions in separate header and source files is not possible with template functions.

[7]

- c) Write C++ code for a template class `MyComplex` class that can be used to instantiate and operate on complex numbers. The class should include:
- A constructor with two generic parameters of the same type to represent the complex number's real and imaginary parts.
 - A friend function that overloads the insertion (`<<`) operator for printing complex numbers.

[10]

3. This question deals with C++ exceptions.

- a) What is "exception safe" code?

[4]

- b) Consider the following function:

```
int min_value(const std::vector<int>& v);
```

Which returns the minimum value contained in the input vector. Discuss why and how the use of exceptions might be appropriate in this case.

[6]

- c) A function contains an `exit` instruction that is executed when something is not right. Discuss why and how the use of exceptions might be appropriate in this case.

[6]

- d) Consider the following code. Trace its execution and its output including an explanation for all the steps.

```
#include <iostream>
#include <string>
#include <exception>

class class_a{
public:

    class_a(int i_n) : n(i_n){
        std::cout << "a" << n << std::endl;
    }

    void f(){
        std::cout << "f1" << n << std::endl;
    }
};
```

```

        std::string s = "z";
        throw s;
        std::cout << "f2" << n << std::endl;
    }

    ~class_a(){
        std::cout << "t" << n << std::endl;
    }

private:
    int n;
};

int main(){
    class_a c1(8);

    try{
        class_a c2(9);
        std::cout << "1" << std::endl;
        c1.f();
        std::cout << "2" << std::endl;
        std::exception e;
        throw e;
        std::cout << "3" << std::endl;
    }
    catch(const std::exception& e){
        std::cout << "e" << std::endl;
    }
    catch(const std::string& msg){
        std::cout << msg << std::endl;
    }

    std::cout << "3" << std::endl;

    return 0;
}

```

[14]

