School of Computing and Information Technology

Student to complete:	
Family name	
Other names	
Student number	
Table number	

CSCI251
Advanced Programming
Hong Kong Campus

Examination Paper
Spring Session 2021 (Online Version)

Exam duration	4 hours
Mode of Exam	Online Mode
Directions to students	- This exam contains 6 questions, for a total of 50 marks. - Download files (if provided) as instructed in questions. - All answers must be coded in a proper IDE like Visual Studio, and XCode, etc. - Compress your files into a .zip file. Rename the .zip file as *fullname_sid*. (E.g. garretlai_12345678.zip) - Submit your .zip file to Moodle dropbox. - Dropbox will be closed sharply 4 hours after commencement of exam.
	This paper is worth 60% of the total marks for the subject.

Q1 Create an MS Word file and name it as <UID>_CSCI251_Q1.docx. Write your answers on the file.

(18 marks in total)

a) State ONE advantage of smart pointers over regular pointers (raw pointers).

(1 mark)

b) Suppose `p` is valid, state ONE scenario that the programme would cause an error if a programmer wishes to remove object `p` from the memory:

```
…
ptr = &p;
delete ptr;
```

(1 mark)

c) Explain, with a scenario, where it is preferable to provide a virtual destructor when designing a class.

(1 mark)

d) Study the following code segment and explain whether the code segment is valid.

```
class Base{
    public:
    virtual void foo() final{
        cout << "foo() from base class" << endl;
    }
};

class Derived : public Base{
    public:
    void foo() override{
        cout << "foo() from derived class" << endl;
    }
};
```

(1 mark)

e) An array name is often considered as a "constant pointer" to an array. Consider the following statements and state whether there is any difference in the byte size of `ptr1` and `ptr2`.

```
int ptr1[5];
int * const ptr2 = new int[5];
```

(1 mark)

f) The following code segment fails to initialize the array elements to 1. State the usage of the keyword `auto`. Rewrite the code segment so that it could work.

```
int *ptr = new int[5]; //Don't modify this line

for(auto& x : ptr)
      x = 1;
```

(2 marks)

g) The following code segment fails to avoid memory leak even with a delete statement. Rewrite the programme so that memory leak could be prevented.

(2 marks)

```
int **ptr = new int*[5];
for (int i = 0; i < 5; i++)
      ptr[i] = new int (i+1);

delete ptr;
```

h) Explain the potential problem of the following code segment.

(1 mark)

```
int& foo(int x){
    int temp = x * 2;
    return temp;
}

int main(){
    int q = foo(5);
    cout << q << endl;
}
```

i) State the problem of the following code segment.

```
#include <iostream>
int main(){
      int *p = new int(3);
      int *q = p;

      delete p;
      p = nullptr;
      std::cout << *q << std::endl;
}
```

(1 mark)

j) Explain whether it is always preferable to set attributes as `protected` over `private` when a programmer knows his/her class will have derived classes.

(1 mark)

k) Below is an overview of the programme structure. Function `outputInfo()` is defined in class A, B, and C, except class D. Instead, D has a function called `extraInfo()`. In `main()`, an iteration loops through the array and invokes `outputInfo()` of the corresponding element. Code several lines of a statement so that `extraInfo()` will be invoked when the element points to an object instance of D.

```
class A{
    ...
    // outputInfo() has been defined here
};

class B : public A{
    ...
    // outputInfo() has been defined here
};

class C : public A{
     ...
     // outputInfo() has been defined here
};

class D : public C{
    void extraInfo() const{
        cout << "...." << endl;
    }
}
int main () {
    A* ls[3];
    ls[0] = new B(...);
    ls[1] = new C(...);
    ls[2] = new D(...);

    for(int i = 0; i < 5; i++){
        ls[i]->outputInfo();
        // Q: how can the programme excute extraInfo() when it is pointing D object?

    }
    ...
}
```

(3 marks)

l) Continue to part k, state the type of casting you have used in part k.

(1 mark)

m) The following programme fails to compile. State the reason and suggest a solution to resolve the situation. Note that these files must be separated.

```
const int roomTemp = 25;
```
RoomTemp.h

```
#include "RoomTemp.h"

int increaseTemp(int temp){
   return roomTemp + temp;
}
```
ComputeTemp.h

```
#include "RoomTemp.h"
#include "ComputeTemp.h"
#include <iostream>
int main(){
   std::cout << "Room temparature: " << roomTemp << endl;
}
```
main.cpp

(2 marks)

Q2 For Q2, use C++ IDE to create corresponding `Q2.h` and `Q2.cpp` file that fulfil the following requirements. Download the `main.cpp` in the Q2 folder. It is for your testing purpose. DO NOT modify the `main.cpp`.

Compress your files in a .zip file and name it as StudentID_CSCI251_Q2.zip.

(10 marks in total)

a) Create a class called `CampEquipment` with the following attributes: `brand`, `rentalFee`, and `name`. `rentalFee` is a pointer to an array with 3 elements.

(1 mark)

b) Create TWO constructors for this class:

i. A default constructor setting attributes: `brand`, `rentalFee`, and `name` as `""`, `{0, 0, 0}`, and `""`, respectively. Display `"Default constructor is invoked."` On screen.

ii. A constructor is taking 3 parameters: `brand`, `rentalFee` pointer, and `name`. Assign these values to the corresponding attributes. Display `"Other constructor is invoked."` On screen.

(2 marks)

c) Create corresponding inspector and mutator methods for the attributes mentioned above.

(2 marks)

d) Explain whether a destructor is necessary. If necessary, create a destructor and display "`Destructor is invoked.`" on the screen. If not, explain the reason by putting comments on the .h file.

Note: Mark will not be given if you have created a destructor where it is not necessary.

(1 mark)

e) Explain whether a user-defined copy constructor is necessary. If it is necessary, create a user-defined copy constructor and display "`Copy constructor is invoked.`" on screen. If not, explain the reason by putting comments on the .h file.

Note: Mark will not be given if you have created a user-defined copy constructor where it is not necessary.

(2 marks)

f) Create a move constructor and display "`Move constructor is invoked.`" on screen.

(2 marks)

Remarks: Data encapsulation should be preserved.

Q3 Download the `main.cpp` in folder Q3. The programme encounters a run-time error if users input 0 as the divisor. Use exception handling method to avoid this issue. Display a message "Invalid! Divisor is zero!".

(3 marks)

Q4 Download the `Q4.cpp` file from folder Q4. Develop `Q4.cpp` according to the following instructions. There are several classes you are asked to create, put these classes on the `Q4.cpp` file.

Note that the following parts serve only as a guide. You may need to re-modify your work so that the programme could work properly.

(12 marks in total)

a) Construct a class called Sport:
   i. Class Sport contains an attribute `sportName`.
   ii. Create necessary constructor(s), inspector(s), mutator(s) and a destructor that make the `main()` method operable.
   iii. Create a method called `outputInfo()` with `void` as the return type. Make this method a pure virtual method.

(3 marks)

b) Construct a class called WaterSport that publicly inherits Sport:
   i. Class WaterSport contains a boolean attribute called `indoor`.
   ii. Create necessary constructor(s), inspector(s), mutator(s) and a destructor that make the `main()` method operable.
   iii. Override the Sport's `void outputInfo()` and print a message "`<sportName> is an <indoor/outdoor> water sport.`" `<sportName>` is the value from the Sport class and `<indoor/outdoor>` depends on the boolean value.

(3 marks)

c) Construct a class called LandSport that publicly inherits Sport:
   i. Class LandSport contains an attribute called `typeOfSurface`;
   ii. Create necessary constructor(s), inspector(s), mutator(s) and a destructor that make the `main()` method operable.
   iii. Override the Sport's `void outputInfo()` and print a message "`<sportName> is a land sport on <typeOfSurface>.`"

(3 marks)

d) Construct a class called Triathlon that publicly inherits both WaterSport and LandSport.
   i. Create necessary constructor(s), inspector(s), mutator(s) and a destructor that make the `main()` method operable.
   ii. Override the `void outputInfo()` and print a message "`Triathlon is both water sport and land sport on.`"

(2 marks)

e) Explain whether a Sport class object instance can be created. Put your answer in the `Q4.cpp` as a comment.

(1 mark)

Q5 Download the `Q5.cpp` in folder Q5. Create a template class `Q5` so that it could handle multiple data types of attributes. Within the class `Q5`, create a default constructor that prints a message "`Constructor Called`".

(3 marks)

Q6 Download the `Q6.cpp` in folder Q6. Develop two operator overloading:
a) Create a **binary operator** + method.
b) Create a **prefix operator** ++ method.

Refer to the expected output in `Q6.cpp` to obtain more information on how these operators should work.

(4 marks)

- END -