UOW AUSTRALIA

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 2022 (Online Version)

Exam duration	3 hours
Mode of Exam	Online (via Zoom)
Directions to students	1. This exam contains 8 questions, for a total of 50 marks. 2. Create a MS Word file (.docx or .doc) with your full name and student id as file name, i.e. ChanTaiMan_H1234567.docx. 3. You have 5 minutes to submit your work when you are told to stop. 4. Moodle dropbox will close sharply 5 minutes after end of examination. 5. You must keep your camera on and no discussion is allowed.

Create a MS Word file with your full name and student id as file name, i.e. ChanTaiMan_H12345678.docx and attempt all questions. Use separate page for each questions.

1 Attempt the following questions. Each part carries 1 mark.

a) Differentiate `inline` and `constexpr` functions in terms of their time of evaluation.

b) Explain the importance of `<concepts>` being introduced in C++ 20.

c) List any TWO incidents that a copy constructor is called.

d) Suggest a way if one wishes to return multiple values from a function.

e) Determine which new feature of C++ 20 has been adopted.

```
int main() {

    std::vector<int> num = {1, 2, 3, 4, 5, 6};

    auto res = num | std::views::filter([](int n){ return n % 2 == 0; })
                   | std::views::transform([](int n){ return n * 2; });

    for (auto v: results) std::cout << v << " ";

}
```

2 Adopt the features provided in C++17 <optional> to complete task (a) to (c) according to the instructions stated in the following code segments.

```
#include <iostream>
#include <optional>
using std::string, std::cout, std::endl, std::optional;
class Contact{
    private:
          /*Task (a):
             i.   Create 2 attributes homePhone and
                  mobilePhone
            ii.   Note that every contact has values for
                  these 2 attributes
          */
```

(1 mark)

```
    public:
          // some code segments…

          /* Task (b): Implement getHomePhone()
             i.   return homePhone if it has value or "N/A"
                  otherwise
          */
```

(2 marks)

```
          /* Task (c): Implement getFirstContact()
             i.   return mobilePhone if it has value, return
                  homePhone if mobilePhone does not contain
                  any value, or "N/A" if neither attribute
                  contains a value.
          */
```

(2 marks)

```
};

int main(){
    Contact c1;
    Contact c2 ("1234", "5555");
    cout << c1.getHomePhone() << endl;
    cout << c2.getFirstContact() << endl;
    return 0;
}
```

Sample output:

```
N/A
5555
```

3 Use the features of <tuple> and C++17 structured binding to complete task (a) and (b) according to the instructions stated in the following code segment.

(3 marks)

```
#include <iostream>
#include <tuple>

using namespace std;

class Point{
     private:
          int x, y;

     public:
          // some constructors have been done…

          /*Task (a):
               Implement getCoordinate() that returns a tuple
               that contains values of x and y.
          */
};

int main(){
    Point p(1, 2);

    //Task (b):
    //use C++17 structured binding to get the coordinate of p


    cout << "coordinate: (" << x << ", " << y << ")" << endl;
    return 0;
}
```

Sample Output:

```
coordinate: (1, 2)
```

4 According to the following instructions, develop a C++ class called MyClass.

(a) MyClass has 2 member variables: data (int*) and message (string).

(1 mark)

(b) Implement the following constructors:

i. A parameterized constructor that takes 2 parameter and initializes the member variables.

ii. A copy constructor.

(4 marks)

(c) Implement a user-defined destructor ONLY if the hidden default destructor is not enough to completely free the memory. (TWO marks will be deducted if the user-defined destructor is defined but not necessary.)

(2 marks)

(d) Implement a pair of getter and setter methods for message.

(2 marks)

5 Study the following code segments.

```
class Base1 {
  private:
    int n1B1;

  protected:
    int n2B1;

  public:
    int n3B1;
    Base1(){
      cout << " Base1's constructor called" << endl;
    }
};

class Base2 {
 public:
     Base2()
     { cout << "Base2's constructor called" << endl;  }
};

class Derived: public Base1, public Base2 {
   public:
     Derived()
     {  cout << "Derived's constructor called" << endl;  }
};

int main(){
   Derived d;
   return 0;
}
```

(a) Write the output of the following code segment.

(2 marks)

(b) Explain whether methods in `Derived` class have direct access to member variables `n1B1` and `n2B1` in `Base1`.

(2 marks)

(c) Explain whether object instance `d` can have access to `n3B1` if `Derived` inherits `Base1` in <u>protected</u> mode.

(1 mark)

(d) Suppose there is a `MajorC` class that is inherited by `Base1` and `Base2`. Explain the potential problem of this scenario. Rewrite the code segment(s) to resolve this situation.

Note that you are only required to write the parts which modification is needed.

(3 marks)

6 This question contains multiple parts. Code segment at each part should be considered individually.

(a) Write the output of the following code segment.

```
#include <iostream>
using namespace std;
int main()
{
   int x = -1;
   try {
      cout << "Inside try n" << endl;
      if (x < 0)
      {
         throw x;
         cout << "After throw n" << endl;
      }
   }
   catch (int x ) {
      cout << "Exception Caught n" << endl;
   }

   cout << "After catch n" << endl;
   return 0;
}
```

(2 marks)

(b) Write the output of the following code segment.

```
class Base {};

class Derived: public Base {};
int main()
{
   Derived d;
   try {
       throw d;
   }
   catch(Base b) {
        cout<<"Caught Base Exception";
   }
   catch(Derived d) {
        cout<<"Caught Derived Exception";
   }
   return 0;
}
```

(2 marks)

(c) Write the output of the following code segment.

```
#include <iostream>
using namespace std;

class Test {
public:
  Test() {
       cout << "Constructing an object of Test " << endl;
  }
  ~Test() {
       cout << "Destructing an object of Test "  << endl;
  }
};

int main() {
  try {
    Test t1;
    throw 10;
  } catch(int i) {
    cout << "Caught " << i << endl;
  }
}
```

(3 marks)

(d) State any TWO advantages of using Exception in programming.

(2 marks)

7 Study the following programme segment carefully. Develop the following programme by adding the following overloading operators:

(a) overload + operator that returns a new Box object that has the sum of two Box objects' dimensions.

(3 marks)

(b) overload /= that divides its dimension by factor of x.

(3 marks)

(c) overload "<<" print out the volume of the box. Refer the output for its appropriate format.

(3 marks)

```
class Box {
    private:
      double length;      // Length of a box
      double breadth;     // Breadth of a box
      double height;      // Height of a box

    public:

    // Assume all necessary constructors, destructor, and
    // methods are already defined.

    double getVolume() const{return length*breath*height;}

};

// Main function for the program
int main() {
   Box Box1(4, 4, 4);
   Box Box2(12, 13, 10);
   Box Box3;

   // volume of box 1
   cout << Box1;

   // volume of box 2
   cout << Box2;

   // Add two object as follows:
   Box3 = Box1 + Box2;

   // volume of box 3
   cout << Box3;

   // Downsize Box1 by half
   Box1/=2;
   cout << Box1;

   return 0;
}
```

Output:

```
Volume of box: 64
Volume of box: 1560
Volume of box: 3808
```

8 Differentiate `Foo()` and `Moo()` the ways of handling pointer. State the potential problem, if any, in these functions. State which method is preferred in terms of better memory management.

(2 marks)

```
void Foo(){
    Song* pSong = new Song(…);
    //…
    string s = pSong->duration;
}

void Moo(){
    unique_ptr<Song> song2(new Song(…));
    //…
    string s = song2->duration;
}
```

- END -