FINAL EXAM

S11159005 黃毓峰

1. We reproduce the class Money here, in part:

class Money

{

public:

Money( );  
 Money(int dollars, int cents);

Money(int dollars);  
 Money(double amount);  
// other public members

const Money operator+(const MoneyADD& amt2)ADD;

>>Need to add & and semicolon

int getCents( ) const;

int getDollars( ) const;

private:

int dollars;

int cents;

//other private members

};

Note that \* is not overloaded in the class, but operator + is overloaded using an operator function with the following declaration:

const Money operator+(const Money& amt2);

The question is: Given the declarations,

Money baseAmount(100, 60); // $100.60

Money fullAmount;

which of the following operations are legal? If so, why? If not, why not?

* 1. BaseAmount + 25; 有加法運算子, 且25會自動轉成Money Class, 所以合法

b) 25 + BaseAmount; 有加法運算子, 且25會自動轉成Money Class, 所以合法

c) baseAmount = 2 \* baseAmount; 不合法因為沒有定義乘法運算子

1. baseAmount+baseAmount. 有加法運算子, 所以合法
2. Consider the class definition:

class IntPair  
{  
 int first;  
 int second;  
public:  
 IntPair(int firstValue, int secondValue);  
 const IntPair operator++( ); //Prefix version  
 const IntPair operator++(int); //Postfix version

int getFirst( ) const;

int getSecond( ) const;

};

Is the following legal? Why or why not?

IntPair a(1,2); 有建構子

(a++)++; 改運算子為const

3. Which of the following defines the C-string containing the word "Welcome"?

a) char stringVar[10] = "Welcome"; 有 '\0'

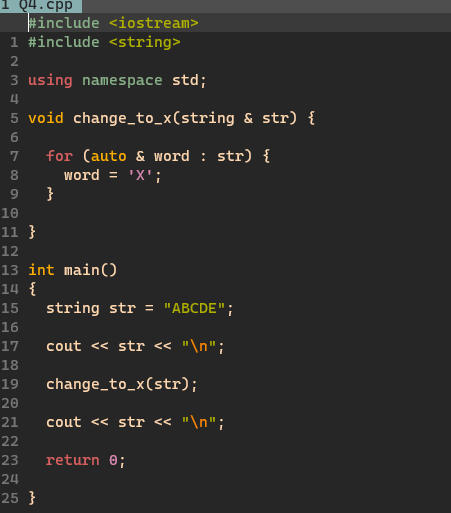
b) char stringVar[10] = {'W', 'e', 'l', 'c', 'o', 'm', 'e'}; 無 '\0'

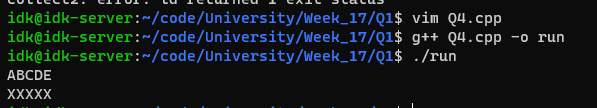
c) char stringVar[10] = {'W', 'e', 'l', 'c', 'o', 'm', 'e', '\0'}; 有 '\0'

d) char stringVar[8] = "Welcome"; 有 '\0'

e) char stringVar[] = "Welcome"; 有 '\0'

4. Given the following definition and initialization. Write a code fragment including a loop that will overwrite the string greeting, with ‘X’ in all character position, keeping the length the same.





5. Give the output from this code fragment:

int \*p1, \*p2;

p1 = new int;

p2 = new int;

\*p1 = 10;

\*p2 = 20;

cout << \*p1 << “ “ << \*p2 << endl;

\*p1 = \*p2;

cout << \*p1 << “ “ << \*p2 << endl;

\*p1 = 30;

cout << \*p1 << “ “ << \*p2 << endl;

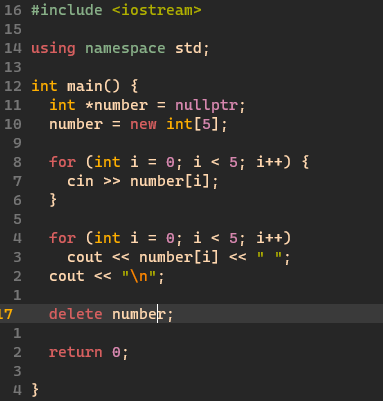
10 20

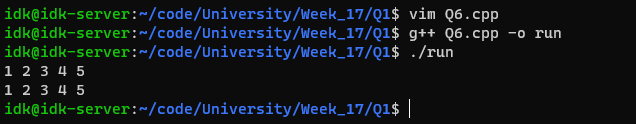
20 20

30 20

6. Create a dynamically allocated array called numbers where the pointer variable numbers points to the array. Write code to input 5 integers from the keyboard and store them in this dynamically allocated array.

Please make sure to include proper input handling and memory deallocation in your code.





7. Here is a list of file names with extensions. Pick at least one that could properly be the implementation file and at least one that can properly serve as interface file. and why?

* 1. File.cxx
  2. File.hxx
  3. File.cc
  4. File.h
  5. File.cpp
  6. File.hpp

Implementation file:

a) File.cxx

c) File.cc

e) File.cpp

Interface file:

b) File.hxx

d) File.h

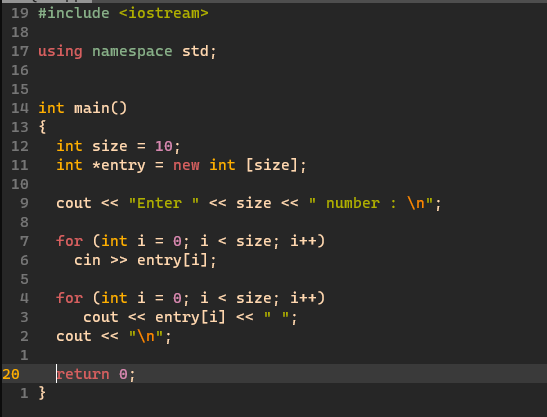
f) File.hpp

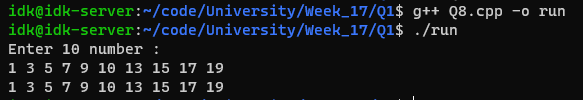
8. Your program creates a dynamically allocated array as follows:

int \*entry;

entry = new int[10];

so that the pointer variable entry is pointing to the dynamically allocated array. Write code to fill this array with 10 numbers typed in at the keyboard.





9. Which of the following sets of statements will set floating point output to the stream outStream to fixed point with set 3 places of decimals? In the explanation, you must give any necessary #include directives and using directives or declarations.

* 1. outStream.setf(ios::fixed);

outStream.setf(ios::showpoint);

outStream.precision(2);

* 1. outStream.setf(ios::fixed | ios::showpoint);

outStream << setprecision(2);

* 1. outStream << setflag(ios::fixed);

outStream << setflag(ios::showpoint);

outStream << setprecision(2);

* 1. outStream.flags(ios::fixed);

outStream.flags(ios::showpoint);

outStream.precision(2);

都只會到小數點後兩位



10. How many times is the following code invoked by the call recursive(4)?

void recursive( int i )

{

using namespace std;

if (i < 8)

{

cout << i << " ";

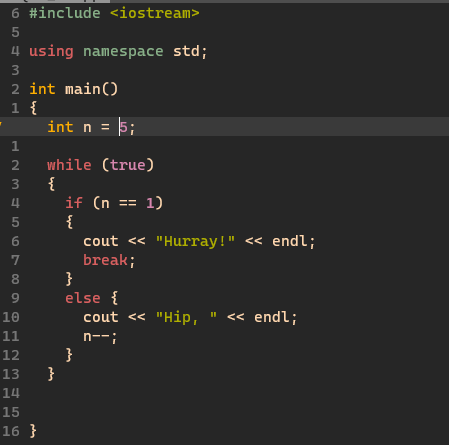
recursive(i);

}

}

* 1. 2
  2. 4
  3. 8
  4. 32
  5. **This is an infinite recursion.**

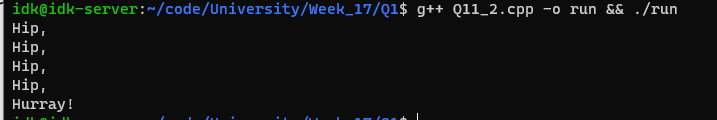
11. Here is a recursive function. Write an iterative version of it. Hint: Be sure you get the number of copies of "Hip" right.

void rec\_cheers(int n)

{  
 using namespace std;  
 if(1==n)  
 cout << "Hurray!" << endl;  
 else  
 {  
 cout << "Hip, ";  
 rec\_cheers(n-1);

}

}



12. Consider the class inheritance.

class B

{

public:

B();

B(int nn);

void f();

void g();

private:

int n;

};

class D: public B

{

public:

D(int nn, float dd);

void h();

private:

double d;

};

How many private members does an object of class D have?

* 1. 0
  2. 1
  3. 2 Private : { method : 0, object : 2 }
  4. 3
  5. 4
  6. 5