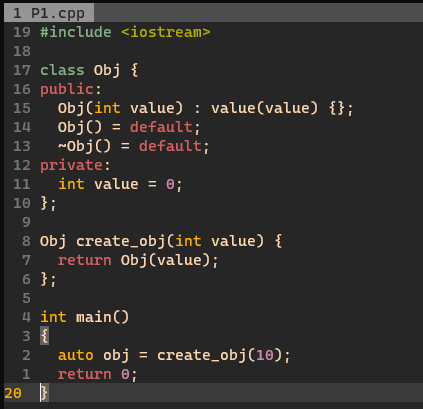
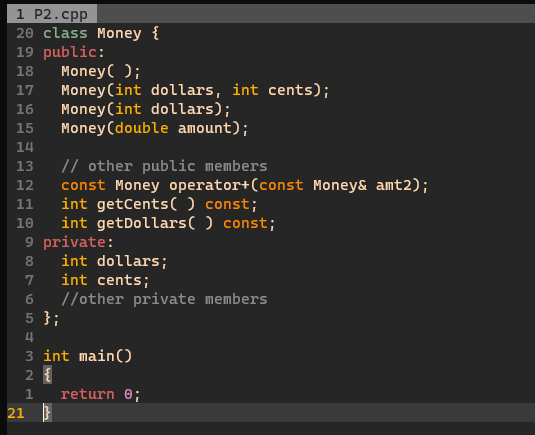
S11159005 黃毓峰

Overloading

1. If I need to build an object for return from a function, I can construct that function directly in the return statement. show a simple example.



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



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?

a) BaseAmount + 25; Legal, 25 will be a Money type

b) 25 + BaseAmount; Legal, 25 will be a Money type

c) baseAmount = 2 \* baseAmount; Not Legal, Money not have \* operator

d) baseAmount+baseAmount. Legal, Money have \* operator

1. A function can return a value of type T in one of several different ways. Which of these is correct? If correct, give a short declaration of a function that returns its value by these mechanism. Comment on the const methods of function return value. Comment on when there is danger involved in returning a reference of any kind from a function. Finally, which one of these returns an l-value and which an r-value?
   1. By value T f()
   2. By lazy evaluation. Not Define

c)      By reference T & f ()

d)      By const value const T f()

e)      By const reference const T & f()

String

1. Which of the following defines the C-string containing “Hello”?

a) char stringVar[10] = “Hello”;

True – the total len is 6, because have ‘\0’

b) char stringVar[10] = {‘H’, ‘e’, ‘l’, ‘l’, ‘o’};

False – they not have ‘\0’

c) char stringVar[10] = {‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’};

True – have ‘\0’

d) char stringVar[6] = “Hello”;

True – have ‘\0’, and fit the array.

e) char stringVar[] = “Hello”;

True – have ‘\0’, and right value is const, so compiler know the size.

1. If you have already removed a character (in char variable ch) from the input stream, you can apply a member function of cin to the variable ch to return it to the input stream. Which of these does this task? given a explanation.

a) cin.get(ch); not we want – the method is get the value from the istream.

b) cin.put(ch); we want – the method is put the variable to the istream.

c) cin.peek(ch); not we want – the method get what value on the istream,   
the value is still on the istream.

d) cin.putback(ch); we want – the method is put back the variable to the istream.

1. There are several uses of the getline function. Assume the definition, string line1, line2;  Which one of these allows specification of the character on which input stop?

a)      cout << “Enter a line of input \n”;

getline(cin, line1);

unit read the ‘\n’ or EOF

b)       cout << “Enter a line of input \n”;

getline(cin, line1, ‘?’);

read unit ‘?

c)      cout << “Enter a line of input \n”;

getline(cin, line1) >> line2;

not legal

1. On most systems, you don’t get a chance to enter the letter when the code given below is run. Write two different code fragments that repair the problem.

cout << “Enter a number:\n”;

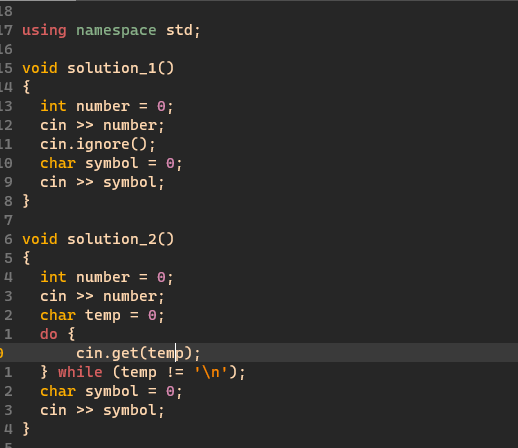
int number;

cin >> number;

cout << “Enter a letter;\n”;

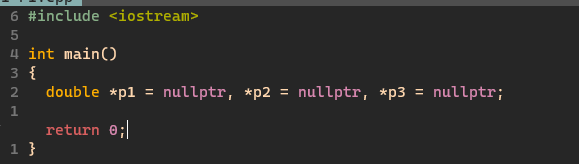
char symbol;

cin.get(symbol);  
 cout << number << “ “ << symbol << endl;



Pointers and Dynamic Arrays

1. The declaration below declares three pointer variables of type pointer to double that is,  a pointer of type (double\*)



1. Give the output from this code fragment:

int \*p1, \*p2;

p1 = new int;

p2 = new int;

\*p1 = 10;

\*p2 = 20;

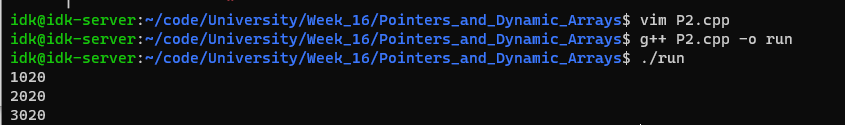
cout << \*p1 << “ “ << \*p2 << endl;

\*p1 = \*p2;

cout << \*p1 << “ “ << \*p2 << endl;

\*p1 = 30;

cout << \*p1 << “ “ << \*p2 << endl;



Separate Compilation and Namespaces

1. Which of the following are the correct preprocessor commands necessary to prevent multiple inclusions of header files? If there is an answer here that works to prevent multiple inclusion, carefully describe how it works in the explanation.
   1. #include “header.h”

If is from .cpp file than its true

b)      #define HEADER\_H

#ifndef  HEADER\_H

//declarations for header.h go here

#endif

It is not true, because if exists two source files include the .h file, they will double define it may make an error.

c)      #ifndef  HEADER\_H

#define HEADER\_H

// declarations for header.h go here

#endif

It is true, because it will detect the .h file has been include or not.

d)      #ifndef  HEADER\_H

//declarations for header.h go here

#endif

It is not true, because, it will always include.

1. 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.

|  |  |
| --- | --- |
| a)      File.cxx  b)      File.hxx  c)      File.cc | d)      File.h  e)      File.cpp  f)       File.hpp |

Implementation file:

a) File.cxx

c) File.cc

e) File.cpp

Interface file:

b) File.hxx

d) File.h

f) File.hpp

1. Suppose the following code is embedded in an otherwise correct and complete program. Answer below the question about what version of f()is called in g().

void f(); //in the global namespace

namespace A

{

  void f();

  void g()

  {

f();  //Does this call A::f()? Or the global f()?

  }

}

a) The call is to the global f();

b) The call is to the namespace A version of f(), i.e., A::f();

c) There is an error. There is a conflict between the namespace f() and the global f(), so there is no call made at all

d) There are other errors that prevent the code from running.

Streams and file I/O

1. 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.

a) outStream.setf(ios::fixed);

outStream.setf(ios::showpoint);

outStream.precision(2);

b)      outStream.setf(ios::fixed | ios::showpoint);

outStream << setprecision(2);

c) outStream << setflag(ios::fixed);

outStream << setflag(ios::showpoint);

outStream << setprecision(2);

d) outStream.flags(ios::fixed);

outStream.flags(ios::showpoint);

outStream.precision(2);

都只會有小數點後兩位



1. The \_\_\_\_\_\_\_\_\_\_\_\_  fstream member function opens a file stream and connects the stream variable to a physical file whose name is the argument to the function.

a)      close( )

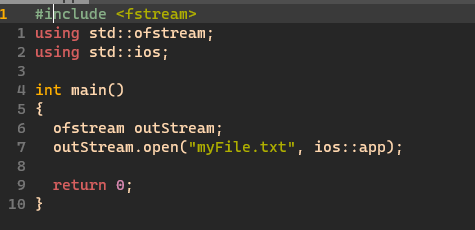
b)      overloaded operator <<( )

c)      open( )

d)      eof( )

e)      flush( )

1. You have a file that is not empty, and you want to preserve the contents and append to the end of the file. Give the commands necessary to open a file for appending.



Recursion

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

