Faculty of Natural and Applied Sciences

Section A MWF
Section B TTH
Nouhad J. Rizk
Fall 2000

Final Exam CSC 313

Time: 2 hrs
Feb 2, 2001

Name : _____ **Section :** ____

Parts	Grade	Subtotals:
Part I Multiple choice /35		
Part II: Dynamic Structure /25		
Part III Overloading /15		
Part IV: Matrix /25		
Total		

Part I: Multiple Choice

1- Here is the start of a class declaration:

```
class foo
{
public:
  void x(foo f);
  void y(const foo f);
  void z(foo f) const;
```

Which of the three member functions can alter the PRIVATE member variables of the foo object that activates the function?

a. Only x can alter the private member variables of the object that activates the function.
b. Only y can alter the private member variables of the object that activates the function.
c. Only z can alter the private member variables of the object that activates the function.
d. Two of the functions can alter the private member variables of the object that activates the function.
e. All of the functions can alter the private member variables of the object that activates the function.

2- Is it possible for a member function of a class to activate another member function of the same class?

a. No.
b. Yes, but only public member functions.
c. Yes, but only private member functions.
d. Yes, both public and private member functions can be activated within another member function.

3- Can two classes contain member functions with the same name?

a. No.
b. Yes, but only if the two classes have the same name.
c. Yes, but only if the main program does not declare both kinds
d. Yes, this is always allowed.

4- What is the primary purpose of a default constructor?

a. To allow multiple classes to be used in a single program.
b. To copy an actual argument to a function's parameter.
c. To initialize each object as it is declared.
d. To maintain a count of how many objects of a class have been created.

5- Suppose that the foo class does not have an overloaded assignment operator. What happens when an assignment a=b; is given for two foo objects?

a. The automatic assignment operator is used.
b. The copy constructor is used
c. Compiler error
d. Run-time error

6- When should you use a const reference parameter?

a. Whenever the data type might be many bytes.
b. Whenever the data type might be many bytes, the function changes the parameter within its body, and you do NOT want these changes to alter the actual argument.
c. Whenever the data type might be many bytes, the function changes the parameter within its body, and you DO want these changes to alter the actual argument.
d. Whenever the data type might be many bytes, and the function does not change the parameter within its body.

7- Here is a small function definition:

```
void f(int i, int &k)
{
  i = 1;
  k = 2;
}
```

Suppose that a main program has two integer variables x and y, which are given the value 0. Then the main program calls f(x,y); What are the values of x and y after the function f finishes?

a. Both x and y are still 0.
b. x is now 1, but y is still 0.
c. x is still 0, but y is now 2.
d. x is now 1, and y is now 2.

8- Here is a function prototype and some possible function calls:

```
int day_of_week(int year, int month = 1, int day = 1);
// Possible function calls:
  cout << day_of_week( );
  cout << day_of_week(1995);
  cout << day_of_week(1995, 10);
  cout << day_of_week(1995, 10, 4);
```

How many of the function calls are legal?

a. None of them are legal
b. 1 of them is legal
c. 2 of them are legal
d. 3 of them are legal
e. All of them are legal

9- Which kind of functions can access private member variables of a class?

a. Friend functions of the class
b. Private member functions of the class
c. Public member functions of the class
d. All of the above can access private member variables
e. None of the above

10- Suppose cursor points to a node in a linked list (using the node definition with member functions called data and link). What statement changes cursor so that it points to the next node?

a. cursor++;
b. cursor = link( );
c. cursor += link( );
d. cursor = cursor->link( );

11- Suppose cursor points to a node in a linked list (using the node definition with member functions called data and link). What Boolean expression will be true when cursor points to the tail node of the list?

a. (cursor == NULL)
b. (cursor->link( ) == NULL)
c. (cursor->data( ) == NULL)
d. (cursor->data( ) == 0.0)
e. None of the above.

12-. Why does our node class have two versions of the link member function?

a. One is public, the other is private.
b. One is to use with a const pointer, the other with a regular pointer.
c. One returns the forward link, the other returns the backward link.
d. One returns the data, the other returns a pointer to the next node.

13- Suppose that p is a pointer variable that contains the NULL pointer. What happens if your program tries to read or write *p?

a. A syntax error always occurs at compilation time.
b. A run-time error always occurs when *p is evaluated.
c. A run-time error always occurs when the program finishes.
d. The results are unpredictable.

14- Suppose that f is a function with a prototype like this:

```
void f(_____ head_ptr);
// Precondition: head_ptr is a head pointer for a linked list.
// Postcondition: The function f has done some computation with
// the linked list, but the list itself is unchanged.
```

What is the best data type for head_ptr in this function?

a. node
b. const node
c. node*
d. const node*

15- What is the output of these statements, using your sequence ADT implemented as a linked list with Item defined as integer:

```
sequence x;
sequence y;
x.insert(41);   // Inserts 41 into the sequence x
x.insert(42);   // Inserts 42, so that x is now 42, 41 with  cursor at front
y = x;
x.attach(43);   // Attaches 43 so that x is now 42, 43, 41 with cursor at 43
y.advance( );
cout << "y size is " << y.size( );
cout << " and y current item is " << y.current( ) << endl;
```

a. y size is 2 and y current item is 41.
b. y size is 2 and y current item is 43.
c. y size is 3 and y current item is 41.
d. y size is 3 and y current item is 43.
e. None of the above.

16- Suppose that you forgot to override the assignment operator in your sequence ADT implemented as a linked list. What is the most likely output from the previous question?

a. y size is 2 and y current item is 41.
b. y size is 2 and y current item is 43.
c. y size is 3 and y current item is 41.
d. y size is 3 and y current item is 43.
e. None of the above.

17- What kind of list is best to answer questions such as "What is the item at position n?"

a. Lists implemented with an array.
b. Doubly-linked lists.
c. Singly-linked lists.
d. Doubly-linked or singly-linked lists are equally best

18- What is the primary purpose of template functions?

a. To allow a single function to be used with varying types of arguments
b. To hide the name of the function from the linker (preventing duplicate symbols)
c. To implement container classes
d. To permit the use of the debugger without the -gstabs flag

19- Consider this prototype for a template function:

```
template <class Item>
void foo(Item x);
```

Which is the right way to call the foo function with an integer argument i?

a. foo(i);
b. foo<int>(i);
c. foo<Item>(i);
d. foo(<int> i);
e. foo(<Item> i);

20- Consider the following definition:

```
template <class Item>
Item maximal (Item a, Item b)
{
   if (a > b)
      return a;
   else
      return b;
}
```

What restrictions are placed on the Item data type for a program that uses the maximal function?

a. The Item data type must be either int, double, or float.
b. The Item data type must be one of the built-in C++ data types.
c. The Item data type must have a copy constructor and a > operator defined.
d. None of the above restrictions apply.

21- When should a function be implemented as a template function?

a. When the data types of the parameters all have copy constructors.
(b) When the function depends on an underlying data type.
c. When the function is relatively short (usually just one line).
d. When the function only takes one argument.

22- When you write a template class, where does the template prefix occur?

a. Before the template class definition
b. Before each member function implementation.
c. Before any other template functions that manipulate the template class.
d. TWO of the above answers are correct.
(e) All of the (a), (b), and (c) are correct.

23- Which of the following stack operations could result in stack underflow?

a. is_empty
(b) pop
c. push
d. Two or more of the above answers.

24- Which of the following applications may use a stack?

a. A parentheses balancing program.
b. Keeping track of local variables at run time.
c. Syntax analyzer for a compiler.
(d) All of the above.

25- Consider the following pseudocode:

```
declare a stack of characters
while ( there are more characters in the word to read )
{
  read a character
  push the character on the stack
}
while ( the stack is not empty )
{
  write the stack's top character to the screen
  pop a character off the stack
}
```

What is written to the screen for the input "carpets"?

a. serc
b. carpets
(c) steprac
d. ccaarrppeettss

26-Here is an INCORRECT pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while ( more input is available)
{
  read a character
  if ( the character is a '(' )
    push it on the stack
  else if ( the character is a ')' and the stack is not empty )
    pop a character off the stack
  else
    print "unbalanced" and exit
}
print "balanced"
```

Which of these unbalanced sequences does the above code think is balanced?

a. (()
b. ())(
c. (()))
d. ()()(

Throughout this section, A is a class and B is a new class derived from A. Also, we have these variables:

```
A a;
B b;
B b1;
B b2;
```

27- What C++ syntax is used to declare that a class B is derived from class A?

a. class A derives B { ... };
b. class B from A { ... };
c. class B : public A { ... };
d. class B subclass of A { ... };

28- If a class B is derived from A, then which of the following terms describes A?

a. ancestor class.
b. base class.
c. parent class.
d. superclass.
e. All of the above.

29- Using the variable declarations at the top of this section, which of the following assignment statements are legal?

a. a = b;
b. b = a;
c. b1 = b2;

d. Both (a) and (b) are legal, but not (c).
e. Both (a) and (c) are legal, but not (b).
f. Both (b) and (c) are legal, but not (a).

30- Consider the assignment statement a=b; (with the variable declarations at the top of this section). Which answer is true?

a. The assignment statement is illegal.
b. The assignment statement activates the A assignment operator.
c. The assignment statement activates the B assignment operator.
d. The assignment statement activates both A and B assignment operators.

31- Consider the declarations at the top of this section. Suppose there are two functions: f has an argument of type A and g has an argument of type B. Which statement is correct?

a. Both f(b) and g(b) are legal function calls.
b. f(b) is legal, but g(b) is not legal.
c. f(b) is not legal, but g(b) is legal.
d. Neither f(b) nor g(b) is a legal function function call.

32- Consider the declarations at the top of this section. Suppose there are two functions: f has an argument of type A and g has an argument of type B. Which statement is correct?

a. Both f(a) and g(a) are legal function calls.
b. f(a) is legal, but g(a) is not legal.
c. f(a) is not legal, but g(a) is legal.
d. Neither f(a) nor g(a) is a legal function function call.

33- One difference between a queue and a stack is:

a. Queues require dynamic memory, but stacks do not.
b. Stacks require dynamic memory, but queues do not.
c. Queues use two ends of the structure; stacks use only one.
d. Stacks use two ends of the structure, queues use only one.

34- If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?

a. ABCD
b. ABDC
c. DCAB
d. DCBA

35- I have implemented the queue with a linked list, keeping track of a front pointer and a rear pointer. Which of these pointers will change during an insertion into a NONEMPTY queue?