CS 271 Exam 3 version B

O O Programming (New Mexico State University)

88

Name Jacob Espinoza
2017-04-20

CS 271 **Exam 3 - Version B** **Spring 2017**
Total Point Value = 100

This exam is all C++. No electonic devices permitted. Please silence and put away cell phones, tablets, computers, etc.

Things to Avoid on this Exam

- Do not use the C language I/O functions printf( ) and scanf( ). Use cout << for output and cin for input. Use C++ I/O manipulators for formatting output, printing newlines, etc.
- Do not use a global variable in any program.
- Do not write comments in your code.

Multiple choice answers must be entered here:

1. (D) public	11. (B) 2
2. (A) data	12. (B) a pointer...
3. (D) zero	13. (B)
4. (A) setHour	14. (B) overloading
✗ 5. (D) template	15. (D) –
6. (B) ::	✗ 16. (A)
7. (A) constructor w/void	17. (B)
8. (C)	✗ 18. (C) private members
9. (B)	19. (C) member functions
10. (C) setprecision	20. (A) zero

51/60

Multiple Choice (3 points each)

1. Any ___(D)___ member of a class is accessible to all functions in all programs.

```
a. data
b. private
c. protected
```
[✓ d. public]

2. ___(A)- data___ members represent the properties or attributes of a class.

[✓ a. data]
```
b. friend
c. private
d. public
```

3. A default constructor for a class has ___(D) 0___ parameter(s).

```
a. const
b. int
c. one
```
[✓ d. zero]

4. Here is a function prototype:

```
Time & setHour ( int );
```

What is the name of the function?

[✓ a. setHour]
```
b. Time
c. Time &
d. the function name is not shown in the prototype
```

5. A function ___(D) template___ is a combination of the function name and the function's parameters.

```
a. prototype
b. signature
c. stack frame
```
[✓ d. template]

6. What is the symbol for the binary scope resolution operator?

```
a. <<
```
[✓ b. ::]
```
c. >>
d. ?:
e. ->
```

7. Which of the following statements will produce a syntax error?

[✓ a. Declaring a constructor as a void function.]
```
b. Overloading a constructor.
c. Writing an accessor as a const function.
d. Writing default arguments to a constructor.
```

The header file of class Lever contains the following class definition.

```
class Lever {
public:
   void setAngle( double f );
   void setMat ( double a );
   int getMat( ) const;
   double getAngle( ) const;
private:
   double angle;
   int mat;
};
```

8. Assuming that the member function definitions are written in a separate .cpp file, which of the following is a correct implementation of the mutator function for angle?

```
a. double Lever::getAngle(__) {
       return angle;
   }

b. void setAngle( double a ) {
       angle = a;
   }

[✓ c.] void Lever::setAngle( double a ) {
       angle = a;
   }

d. double Lever::getAngle(_) {
       return angle;
   }
```

exactly* the same

9. If we want to be able to use a call to the mutator in a chain of function calls, or nested inside another statement, what is the correct prototype of the function?

```
a. double setAngle( double a );
[✓ b.] Lever & setAngle( double a );
c. void Lever::setAngle( );
d. void setAngle( double a );
```

10. Which I/O manipulator is used to determine the number of digits to the right of the decimal point in a float or double value?

```
a. fixed
b. setfill
[✓ c.] setprecision
d. setw
```

11. Given the following class definition:

```
class Example {
   Example( );
   Example *getLink( );
   void setLink( Example * );
private:
   int data;
   Example * link;
public:
   void setData ( int );
   int getData( );
};
```

How many public member functions does class Example have?

a. 0
[✓ b.] 2
c. 4
d. 5

12. Inside a member function definition, what is "this"?

a. a local variable
[✓ b.] a pointer to the calling object
c. a reference to the calling object
d. an array that contains the class's data members

13. In order to allow a non-member function to access the private data members of a class:

a. a non-member function can never access a class's private data
[✓ b.] the class must grant friend permission to the function
c. the class must overload the function
d. the function must request friend permission from the class

14. Function ________________________________ occurs when two or more functions have the same name but different parameter lists.

a. encapsulation
[✓ b.] overloading
c. overriding
d. prototyping

15. Which of the following symbols is used in UML to denote a private member of a class?

a. >>
b. +
c. *
[✓ d.] -

16. What are the four operators that cannot be overloaded in C++?

[✓ a.] ->   .   ::   [ ]
b. .   .*   ::   ?:
c. ^   ~~--~~   ++   [ ]
d. +=   ~~-=~~   *=   /=

17. Given a class called Time with the following data members:

    int hour;
    int minute;
    int second;

    What is the correct prototype for the stream insertion operator << if we want to grant the operator friend permissions?

    a. `friend istream & operator << (istream &, Time &);`
    [✓ b.] `friend ostream & operator << (ostream &, const Time &);`
    c. `friend ostream & operator << (ostream &, Time &);`
    d. `friend ostream & operator << (ostream, Time);`

18. In a UML class diagram, the middle compartment lists the

    a. data members of the class
    b. ~~name of the class~~
    [✓ c.] private members of the class
    d. public members of the class

19. In C++, the actions or behaviors of a class are also known as

    a. friend functions of the class
    b. instance variables members of the class
    [✓ c.] member functions of the class
    d. public members of the class

20. In C++, what is the maximum number of parameters that a destructor can have?

    [✓ a.] 0
    b. 1
    c. the same number of parameters as in the constructor
    d. there is no limit to the number of parameters

Programming Questions

21. (20 points) Write the header file for a class called PurchaseOrder. Include a preprocessor wrapper. **Date** is a class. You may assume that Date.h is available in the same folder.

PurchaseOrder
- customerName : string - purchaseDate : Date - totalCost : float
«constructor» PurchaseOrder( ) «constructor» PurchaseOrder(n : string, t : float) «friend» interestEstimate(p : PurchaseOrder&, r : float) : float + getCustomername( ) : string + getTotalcost( ) : float + setCustomername(n : string) : PurchaseOrder & + setTotalcost(t : float) : PurchaseOrder &

```
#include <iostream>
#include <string>
#include <iomanip>
#include "Date.h"

#ifndef PURCHASEORDER_H
#define PURCHASEORDER_H
using ——
class PurchaseOrder {
  (move to before Private:)
    Private:
      String customerName;
      Date PurchaseDate;
      float totalCost;
    Public:
      PurchaseOrder ();
      PurchaseOrder (string n, float t);
      String getCustomername();
      float getTotalcost();
      PurchaseOrder & setCustomername (string n);
      PurchaseOrder & setTotalcost (float t);
      friend float interestEstimate (PurchaseOrder& p, float r);
};
#endif
```

friend declarations can be placed anywhere within the class

20/20

20. (18 points) Write the function definition (including the function header and body) of the friend function called **interestEstimate**. (See the UML above for information regarding parameters and return type.) Note that this function is **NOT** a member function of the PurchaseOrder class.

The first parameter is a PurchaseOrder object; the second parameter is the annual interest rate (given as a float, not a percent) that the company charges for a credit line on purchases. The function should calculate and return the amount of interest that would be due at the end of one year. (determined by multiplying total cost of the purchase order by the annual rate of interest).

only in the prototype

```
~~friend~~ float interestEstimate(PurchaseOrder& P, float r){
    float interest;

    interest = P.getTotalcost() * r;
-1
    return interest;

}
```

~~2~~1. (2 points) A PurchaseOrder object conains a Date object as one of its data members. This illustrates the object-oriented programming concept of coencapsulation (?).

17/20