FORM 1(put name, form, and section number on test!!!)

EECS 162 Exam I

Review session WNGR 116 6-8pm

True (A) / False (B) (2 pts)

1. A constructor is a special kind of member function. It is automatically called when an object of that class is declared. T
2. You can write a class that is useful with all its constructors in the private section. F
3. ~~`size` and `capacity` of a `vector` are two names for the same thing.~~ File I/O
4. It is legal to call a constructor as a member function of an object of a class, as in

```
class A {
public:
  A(){}
  A(int x) : xx(x) {}
private:
  int xx;
};
int main() {
  A w;
  w.A(2); // Is this legal?
}
```

F w = A(2);

"has a"

5. A class may not have another class type object as a member. F
6. In deep copy, pointers are followed and data and the pointer structure are duplicated. T
7. We can assign a derived class object to a base class object. T base Vehicle * v = &v1; &b1;
8. ~~Vector assignment will provide a deep copy for you.~~ File I/O
9. A constructor can be called implicitly or explicitly. T
10. A static variable of a class cannot be changed. F shared
11. Consider this operator overloading for class Money:

```
Money operator+(const Money& amt1,const Money& amt2);
```

const F const T T non-member

Is the following expression legal?

```
Money m1(17.99), m2(23.57), m3(15, 22);
(m1 + m2) = m3;
```

12. A friend function has access only to the private members and member functions of the class of which it is a friend.
13. When an operator is overloaded as a member of a class, the first parameter listed in parentheses is the calling object.
14. When overloading an operator, you can change the behavior of the operator, making + do things that feel like multiplication, but this is unwise.
15. The include statement, `#include <file.h>` looks in the system defined directory for the file, file.h.
16. The "Big Three" is required for every class.
17. Constructors are inherited because something has to initialize the inherited variables.

Multiple Choice (3 pts):

18. In a vector, which of the following statements is true?
    a) Indexing vector access is range checked.
    b) The range of legal index values for a vector is 0 to the value of `v.size()-1`
    c) To add a value use the member function `v.push_front( )`
    d) To increase or decrease a vector's size v.new_size(newSize);

19. A constructor
    a) can only be used to initialize
    b) must initialize all member variables
    c) can do anything any other method can do, including returning information
    d) usually initializes all, or most, member variables

20. Suppose `class Child` is derived from `class Parent` that was in turn derived from `class GrandParent`. When we destroy an object of class Child, three destructors are called: i) Child, ii) Parent, iii)GrandParent.. What is the order?
    a) Child, Parent, GrandParent
    b) Parent, GrandParent, Child
    c) GrandParent, Child, Parent
    d) GrandParent, Parent, Child
    e) GrandParent, Child, Parent

21. Given the following Money class:

```
class Money {
public:
  Money( );
  Money(int theDollar, int theCents);
  Money(int theDollars);
  Money(double amount);
  const Money operator+(const Money& amt2);
  int getCents( ) const;
  int getDollars( ) const;
private:
  int dollars;
  int cents;
};
```

Note that `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 is illegal?

a) `baseAmount + 25;`

b) `25 + baseAmount;`

c) `fullAmount = baseAmount + 2;`

d) `baseAmount+baseAmount;`

22. The Big Three does not consist of which of the following?

a) Default constructor

b) Copy constructor

c) Destructor

d) Operator=

23. If a class is named `MyClass`, what must the destructor be named?

a) `Erase`
b) `MyClass`
c) Any name the programmer wishes except the name of the class
d) `~MyClass`
e) None of the above.

24. You have a program with a class that is separated into files. The implementation has been changed. Of the interface file, the implementation file and the application file, which must be recompiled?

a) Only the interface?
b) Only the implementation?
c) Only the application?
d) None of the above?
e) Some of the above?

25. A copy constructor has the same name as the class (let's call it A) and has a parameter that

a) is call-by value of an A object
b) is call-by-reference of another class
c) is call-by-reference of an A class object
d) is call-by-name of an object named ~A.
e) none of these

26. Which statement about separate compilation is incorrect?

a) There is no compelling advantage to separate files.
b) Placing client and class implementations in separate files enhances project development.
c) Separate files for interface and implementation enhance reuse.
d) Separating interface and implementation can save considerable time.

27. Here is a list of file names with extensions. Which could be the interface file?

a) File.c
b) File.cc
c) File.h
d) File.cpp

28. If class B contains a class D object as a member, then an object of class B bears what relationship to class D?

a) A *has-a* relationship.

b) A *fraternal* relationship

c) An *is-a* relationship.

d) There is no relationship here.

29. Which of the following describes defining a class so that the implementation of the members is either not known or at least irrelevant to their use

a) walling up

b) encapsulation

c) inheritance

d) Caging up the data and functions

30. Which of the following is correct syntax to declare C++ `class B` to be a `public` base class for derived `class D`

a) `public base class B: class D {/*…*/};`

b) `class D : public class B {/* … */};`

c) `class D : public B {/* … */};`

d) `class B: public D { };`

e) None of the above

31. Suppose `class D` is derived from `class B`. `class B` has a `public` member function whose declaration is `void f();`, and `class D` has its version of the function, `void f();`. Here is a pointer definition and an access to a member function.

```
B *bPtr = new D;
bPtr->f();
```

Suppose this is embedded in an otherwise correct and complete program. Which version of `f()` will be called?

a) `D::f()`

b) `B::f()`

c) This is illegal. You can't assign a `D` object to a variable of type pointer to `B`.

32. Consider the class inheritance.

```
class B {
public:
  B();
  B(int nn);
private:
  int n;
  void f();
  void g();
};
class D: public B {
public:
  D(int nn, float dd);
  void h();
private:
  double d;
};
```

Which of the following functions can be invoked by an object of class D?

a) `f()`
b) `g()`
c) `h()`
d) none of the above
e) all of the above

33. Consider the above inheritance in problem 32 above. How many `public` members does an object of `class D` have?

a) 0
b) 1
c) 2
d) 3
e) 4

34. Given the class declaration, `class D : public B {/*…*/};` Which of the following is NOT true?

a) `public` members of `B` become `public` members of `D`

b) `protected` members of `B` become `protected` members of `D`

c) `private` members of `B` become `public` members of D

d) `private` members of `B` are inaccessible in D.

e) none of the above

35. Suppose `class Child` is derived from `class Parent` was in turn derived from `class GrandParent`. The class Parent and GrandParent are the

a) Predecessor classes of class Child

b) Forbearer classes of class Child

c) Ancestor classes of class Child

d) Descendant classes of class Child

e) None of the above

36. If a `class` represents an amount of money (in US currency like $9.99), then the amount (like $9.99) could reasonably be stored in

a. A member variable of type `double`.

b. Two member variables of type `int`.

c. A `string` of characters (like "9.99").

d. All of the above

e. None of the above

Why encapsulation!