CS 2255 **Homework #4**

MULTIPLE CHOICE

1. Objects are created from abstract data types that encapsulate _____ and _____ together.
   a. numbers, characters
   b. data, functions
   c. addresses, pointers
   d. integers, floats

2. In OOP terminology, an object's member variables are often called its _____, and its member functions are sometimes referred to as its behaviors, or _____.
   a. values, morals
   b. data, activities
   c. attributes, activities
   d. attributes, methods

3. A C++ class is similar to one of these.
   a. inline function
   b. header file
   c. library function
   d. structure

4. Examples of access specifiers are the keywords:
   a. `near` and `far`
   b. `opened` and `closed`
   c. `private` and `public`
   d. `table` and `row`

5. This is used to protect important data.
   a. `public` access specifier
   b. `private` access specifier
   c. `protect()` member function
   d. class protection operator, `@`

6. Class declarations are usually stored here.
   a. On separate disk volumes
   b. In their own header files
   c. In .cpp files, along with function definitions
   d. Under pseudonyms

7. This directive is used to create an "include guard," which allows a program to be conditionally compiled. This prevents a header file from accidentally being included more than once.
   a. `#include`
   b. `#guard`
   c. `#ifndef`
   d. `#endif`

8. When the body of a member function is defined inside a class declaration, it is said to be
   a. static
   b. globally
   c. inline
   d. conditionally

9. A _____ is a member function that is automatically called when a class object is _____.
   a. destructor, created
   b. constructor, created
   c. static function, deallocated
   d. utility function, declared

10. The constructor function's return type is
    a. `int`
    b. `float`
    c. structure pointer
    d. None of these

11. The destructor function's return type is:
    a. `tilde`
    b. `int`
    c. `float`
    d. nothing. Destructors have no return type.

12. When a constructor function accepts no arguments, or does not have to accept arguments because of default arguments, it is called a(n):
    a. empty constructor
    b. default constructor
    c. stand-alone function
    d. arbitrator function

13. This type of member function may be called from a statement outside the class.
    a. public
    b. private
    c. undeclared
    d. global

14. If you do not declare an access specification, the default for members of a class is
    a. `inline`
    b. `private`
    c. `public`
    d. `global`

15. In a procedural program, you typically have _____stored in a collection of variables, and a set of _____ that perform operations on the data.
    a. numbers, arguments
    b. parameters, arguments
    c. strings, operators
    d. data, functions

16. A class is a(n) _____ that is defined by the programmer.
    a. data type
    b. function
    c. method
    d. attribute

17. Members of a class object are accessed with the
    a. dot operator.
    b. `cin` object.
    c. extraction operator.
    d. stream insertion operator.

18. Assuming that `Rectangle` is a class name, the statement

```
Rectangle *BoxPtr;
```

a. declares an object of class `Rectangle`
b. assigns the value of `*BoxPtr` to the object `Rectangle`
c. defines a `Rectangle` pointer variable called `BoxPtr`
d. is illegal in C++

19. When you dereference an object pointer, use the
a. `->` operator
b. `<>` operator
c. dot operator
d. `&` operator

20. This type of member function may be called only from a function that is a member of the same class.
a. public
b. private
c. global
d. local

21. The constructor function always has the same name as
a. the first private data member
b. the first public data member
c. the class
d. the first object of the class

22. This is automatically called when an object is destroyed.
a. constructor function
b. specification deallocator
c. destructor function
d. coroner function

23. A class may have this many default constructor(s).
a. only one
b. more than one
c. a maximum of two
d. any number of

24. Objects in an array are accessed with _____, just like any other data type in an array.
a. subscripts
b. parentheses
c. `#include` statements
d. output format manipulators

25. The process of object-oriented analysis can be viewed as the following steps:
a. Identify objects, then define objects' attributes, behaviors, and relationships
b. Define data members and member functions, then assign a class name
c. Declare private and public variables, prototype functions, then write code
d. Write the `main()` function, then determine which classes are needed

26. Assume that `myCar` is an instance of the `Car` class, and that the `Car` class has a member function named `accelerate`. Which of the following is a valid call to the `accelerate` member function?

a. `Car->accelerate();`
b. `myCar::accelerate();`
c. `myCar.accelerate();`
d. `myCar:accelerate();`

27. If a local variable and a global variable have the same name within the same program, the _____ resolution operator must be used.
    a. variable
    b. ambiguity
    c. scope
    d. global

28. For the following code, which statement is *not* true?

```
class Point
{
   private:
      double y;
      double z;
   public:
      double x;
};
```

    a. `x` is available to code that is written outside the class.
    b. The name of the class is `Point`.
    c. `x`, `y`, and `z` are called members of the class.
    d. `z` is available to code that is written outside the class.

29. What is the output of the following program?

```
#include <iostream>
using namespace std;

class TestClass
{
   public:
      TestClass(int x)
      { cout << x << endl; }

      TestClass()
      { cout << "Hello!" << endl; }
};

int main()
{
   TestClass test;
   return 0;
}
```

    a. The program runs, but with no output.
    b. 0
    c. Hello!
    d. The program will not compile.

30. When a member function is defined outside of the class declaration, the function name must be qualified with the:
    a. class name, followed by a semicolon
    b. class name, followed by the scope resolution operator
    c. name of the first object
    d. `private` access specifier
    e. None of these

31. What is the output of the following program?

```
#include <iostream>
using namespace std;

class TestClass
{
   public:
      TestClass(int x)
      { cout << x << endl; }

      TestClass()
      { cout << "Hello!" << endl; }
};

int main()
{
   TestClass test(77);
   return 0;
}
```

a. The program runs, but with no output.
b. 77
c. Hello!
d. The program will not compile.

32. What is the output of the following program?

```
#include <iostream>
using namespace std;

class TestClass
{
   private:
      int val;
      void showVal()
      { cout << val << endl; }

   public:
      TestClass(int x)
      { val = x; }
};

int main()
{
   TestClass test(77);
   test.showVal();
   return 0;
}
```

a. The program runs, but with no output.
b. 77
c. 0
d. The program will not compile.

TRUE/FALSE

1. True/False: Whereas object-oriented programming centers around the object, procedural programming centers around functions.
ANS: T

2. True/False: Class objects can be defined prior to the class declaration.
ANS: F

3. True/False: The constructor function may not accept arguments.

ANS: F

4. True/False: A destructor function can have zero to many parameters.
   ANS: F

5. True/False: More than one constructor function may be defined for a class.
   ANS: T

6. True/False: More than one destructor function may be defined for a class.
   ANS: F

7. True/False: Object-oriented programming is centered around the object, which encapsulate together both the data and the functions that operate on the data.
   ANS: T

8. True/False: You must declare all data members of a class before you declare member functions.
   ANS: F

9. True/False: You must use the `private` access specification for all data members of a class.
   ANS: F

10. True/False: A private member function is useful for tasks that are internal to the class, but is not directly called by statements outside the class.
    ANS: T

11. True/False: If you do not declare a destructor function, the compiler will furnish one automatically.
    ANS: T

12. True/False: When an object is defined without an argument list for its constructor, the compiler automatically calls the object's default constructor.
    ANS: T

13. True/False: One purpose that constructor functions are often used for is to allocate memory that will be needed by the object.
    ANS: T

14. True/False: One purpose that destructor functions are often used for is to free memory that was allocated by the object.
    ANS: T