CMSC 202 Midterm March 17, 2005

Name: ______________________________ **SSN:** ________________

UserID: ______________________________

(Circle your section)

Section:	**101** – Tuesday 11:30	**102** – Thursday 11:30
	103 – Tuesday 12:30	**104** – Thursday 12:30
	105 – Tuesday 1:30	**106** – Thursday 1:30

Directions

- This is a closed-book, closed-note, closed-neighbor exam.
- Read through the entire test before you begin.
- Start with the questions that are easiest for you. If you have time at the end, come back to the more challenging ones.
- Write CLEARLY, if I cannot read your writing, you will receive a zero for the problem in question.
- Feel free to continue your answer on the backs of the pages, but make sure that you indicate where your answer continues.
- When you are done, read over your answers and then bring your exam to the front of the room.
- **You will need your Picture ID to hand in your exam.**

Score

Page Number	Points Possible	Points Earned
2	10	
3	20	
4	15	
5	16	
6	15	
7	12	
8	12	
TOTAL	**100**	

True/False (10 pts, 1 pts each)

Decide if the following are true (T) or false (F), put the appropriate letter in the blank.

__________ 1. `cout` is used with the extraction operator to print values to standard output

__________ 2. C++ supports the `boolean` data type, but C does not.

__________ 3. The following line of code correctly opens a file named "data.txt":

```
ifstream fin("data.txt");
```

__________ 4. The `iofstream` header file is used for input and output file streams.

__________ 5. When passing command line arguments to your program, argc indicates the index of the last item in the argv array.

__________ 6. Static data members are accessible from all class methods but are only modifiable from static methods.

__________ 7. Static methods can be called without instantiating an object of that type.

__________ 8. operator<< cannot be a member function and must be declared as a friend function.

__________ 9. Private data members can only be accessed by methods of the class.

__________ 10. The following code prints: `100 10`.

```
#include <iostream>

using namespace std;

int main()
{
  int i = 10;
  {
    int i = 100;
    cout << i << " " << ::i << endl;
  }
  return 0;
}
```

_________ pts

Short Answer

The following questions are all related and deal with the same system. Assume that the proper header files have been included.

11. (2 pts) Assume that the **command line** has been passed a single argument, a **filename**. Store the filename in a **C++ string**. Use this **string** to open the file for **writing**.

12. (2 pts) Declare a **vector** of **integers**.

13. (5 pts) Use a **loop** to **prompt** and **read** in **integers** from standard input until a **negative** number is read. **Add** them to the **vector** (except for the negative number).

14. (5 pts) Write code to find the **average** of all of the integers in the **vector**. Use vector **methods** whenever possible.

15. (2 pts) Use <u>**two**</u> different methods to **print** the **7th item** in the **vector** to the standard output stream.

16. (2 pts) In <u>**one line of code**</u>, **remove** the **third item** of the **vector** using one or more vector class **methods**.

17. (2 pts) Write the **prototype** for a function that will **sort** the items in the **vector**.

Class Construction

The following questions all have to do with the same system. Make appropriate decisions about data types, return types, const, and parameter passing.

18. (15 pts) Design a class to represent a **Pot of Gold**. Write only the class **declaration** here, do not implement the methods (yet!). Our application is concerned mostly with the **value** and **portability** of each Pot of Gold. Your class must have:
    a. A **default** constructor
    b. A **non-default** constructor
    c. **<u>2 data members</u>** that represent the **value** and **weight**
    d. Appropriate **accessors** for each data member
    e. Appropriate **mutators** for each data member
    f. A **facilitator** method that calculates the **value-density** of the Pot of Gold (dollars per pound)
    g. An overloaded **addition** operator that will add two Pots of Gold
    h. A **data member** that represents the largest a Pot of Gold can be (100 lbs), all Pots have the same maximum weight.

19. (4 pts) Implement the **non-default** constructor for your Pot of Gold class, use other class methods when appropriate.

20. (4 pts) Implement the **mutator** for your **weight** data member, include code to verify the new value is within appropriate limits.

21. (4 pts) Implement the **value-density** facilitator.

22. (4 pts) Implement the overloaded **addition** operator for your Pot of Gold class

23. (15 pts) Declare a **Leprechaun** class (again, do not implement, yet). Your Leprechaun class must have the following:
    a. A **default** constructor
    b. A **non-default** constructor that accepts (at least) a Pot of Gold that the Leprechaun starts with
    c. **3 data members** that represent the Leprechaun's **height**, **weight**, and his **Pot of Gold**
    d. **Accessors** for each data member
    e. **Mutators** for each data member
    f. Overloaded **<< operator**

_________ pts

24. (4 pts) Implement the **non-default** constructor for your Leprechaun class.

25. (8 pts) Implement the overloaded **<< operator** so that it pushes the following **three** lines to the stream:

```
They're always after me lucky charms!
My Pot of Gold is          xxx.xx lbs.
My Pot of Gold is          $xx.xx
```

Note: the values have exactly **<u>2 points of precision</u>** and the decimals are **vertically** aligned. (For output purposes ONLY) You can assume that the **weight** is no more than **100 lbs**, but the **value** may be up to **$1,000,000.00** (1 million).

26. (4 pts) Describe **two** ways to check for **EOF** when reading a **file**.

27. (4 pts) Explain how "**call by reference**" in C++ is **similar** and **different** than passing **parameters with pointers** in C.

28. (4 pts) Explain **three** ways in which **functions** might handle **unsatisfied PreConditions**.