

University of Maryland  
Baltimore County

CMSC 202 – Computer Science II

Spring 2006

Mid-Term Exam

Sections 0201 – 0205

Lecture Hours: Monday – Wednesday 5:30 PM – 6:45 PM

Exam Date: Wednesday 3/15/06

Exam Duration: 5:30 PM – 6:45 PM

Instructor: Sa'ad Raouf

<b>Name:</b>	
<b>SSN:</b>	
<b>Score:</b>	
<b>Section:</b> (Circle your Section Number below)	

Section	TA	Day	Time	Room
0201	Yong	Monday	7:30 - 8:20pm	ECS 104 A
0202	Yong	Wednesday	7:30 - 8:20pm	ECS 104 A
0203	Jim	Wednesday	11:00 - 11:50am	ECS 104 A
0204	Yong	Monday	2:00 - 2:50pm	ECS 104 A
0205	Yong	Wednesday	2:00 - 2:50pm	ECS 104 A

**Notes:**

1. This exam is a closed book, and a closed notes exam.
2. All answers are to be written on the enclosed exam sheets. Scratch sheets are not allowed. If necessary, you can use the back of the exam sheets.
3. You will need to present your Photo ID when handing in the exam. No exceptions.
4. Please hand in your exam with your section number circled. If your section number is not circled, your exam will not be graded.

1) (2 points each) Write True or False in the TRUE/FALSE column:

		TRUE/FALSE
a)	It does not make a difference if you use the at or [] to access an item in a vector	
b)	A vector supports both a push_back() as well as a push_front	
c)	Overloading the operator+ as a non member function requires 2 arguments.	
d)	Using setw will truncate the data if the field width exceeds the width field	
e)	Return a void when overloading the equality operator ==	
f)	Multiple private: and public: sections are allowed by the compiler in a class definition	
g)	The scope resolution operator is used to qualify a member function	
h)	A constructor is like any other class member function, and can return any type.	
i)	Static data members of a class are created when the first instance of the class is constructed.	
j)	By default, the compiler allows you to assign one vector to another vector	

- 2) (4 points) What do calls to `exit(n)` do?
- i) The program is stopped, and the argument is discarded.
  - ii) The `exit(n)` function is obsolete, and is not supported in C++
  - iii) **The `exit(n)` function stops the program, and the value `n` is passed back to the operating system**
  - iv) The `exit(n)` function temporarily stops the program, and the value `n` is passed back to the operating system. Based on the value `n`, the operating system will determine if the program should restart again from the temporary stopping point
  - v) The `exit(n)` allows the Unix system operator to use the emergency exit, in the event of a fire
- 3) (4 points) Identify the `fstream` member function that opens a file stream, and connects the stream variable to a physical file whose name is the argument to the function?
- i) `close()`
  - ii) overload operator `>>`
  - iii) **`open()`**
  - iv) `eof()`
  - v) `flush()`
  - vi) None of the above
- 4) (4 points) Identify the member function that reads in a single line of input
- i) `Newline()`
  - ii) `Put()`
  - iii) `Get()`
  - iv) `getLine()`
  - v) **None of the above**
- 5) (4 points) Write the pre-processor commands necessary to prevent multiple inclusions for a class header file, assume the name of the class is `Student`, and the name of the header file is `Student.h`?

```
#ifndef STUDENT_H  
#define STUDENT_H  
Class {};  
#endif
```

- 6) (4 points) List 3 issues where vectors are superior to arrays. Give an example for each issue. Use the array declaration `int myArray[10];` for your examples.

**See web site**

- 7) (4 points) Show the output from the following code snippet
- What will the value of `i` be after the for loop has completed?
  - What will the size of the vector be after the for loop has completed?

```
vector < int > integers (5);  
for (unsigned int i = 0; i < integers.size( ); i++)  
{  
    cout << integers.at( i ) << endl;  
    integers.push_back(i);  
}
```

**Answer:**

***Infinite loop***

- 8) (4 points) A function of type void can have?
- i) **No arguments**
  - ii) **Any number of arguments**
  - iii) *No more than 3 arguments*
  - iv) **Arguments are determined in the prototype**
  - v) **Exactly one argument**
  - vi)
- 9) (4 points) A definition of a variable declared outside of a function is?
- i) *Local function variable*
  - ii) **Global variable**
  - iii) *Global function header*
  - iv) *Global function definition*
  - v) *Local variable*

10)(4 points) A local variable declared within a function as static has the following attribute:?

- i) **Retains the last value set, during multiple invocations of the functions**
- ii) Resets the value to the initial value, with multiple invocations
- iii) Can only be initialized to zero
- iv) Is shared among all instances of a class

11)(4 points) A class member that is shared among all objects of a class is called?

- i) Answer:

**static**

12)(4 points) In a vector, which of the following are true:

- i) Indexing a vector using [] is range checked
- ii) **The range of legal index values are 0 to vector.size() - 1**
- iii) **To add a member to a vector, use push\_back**
- iv) To resize a vector, use vector.new\_size(newSize)
- v) To determine if the vector is empty, use vector.is\_empty()

Answer:

13)(4 points) Which of the following are correct comparison between call-by-value, and call-by-reference

- i) Both methods protect against changing the caller's arguments
- ii) Both are memory efficient for all sizes of arguments
- iii) Both methods make a copy of the arguments
- iv) **Call-by-value uses more memory**

v) Answer:

14)(4 points) Which of the following are true when using aggregation

- i) **You can define a class within a class**
- ii) The inner class can only be private
- iii) Only the inner class default constructor can be called by the outer class constructor
- iv) **The inner class can be static**

v) Answer:

15)(4 points) Which of the following are true regarding constructors (assume the class name is C)

i) **A default constructor must be explicitly defined for the following to statement to be syntactically valid**

(1) **C x;**

ii) To call the default constructor use the following to declare an object x of type C:

(1) **C x(a = 0);**

iii) **Even though a constructor is a member function, it may not be called in the same way that member functions are called.**

iv) **A constructor has the same name as the class**

v) A constructor can return a false value, to indicate that an invalid initialization value was passed to the constructor

vi) **Answer:**

16)(4 points) A non member function that can have access to the private data members of a class are declared as what type of functions?

a) **Answer:**

**friend**

17)(6 points) Write the function prototype to overload the operator+ for a class called C as:

- i) A friend function
- ii) A member function
- iii) A non-friend, non-member function

**Answer:**

18)(8 points) Explain each of the following declarations

- a) `vector <int> a[10];`
- b) `vector <int> b(10);`
- c) `vector <int> c;`
- d) `int d [10];`
- e) `int e (10);`
- f) `int f;`
- g) `int g = 10;`
- h) `vector <int * > g;`

19)(10 points) Rewrite the following class definition, using `const` and reference to `const` where applicable. Assume that the convention of a member function name starting with **set** is a mutator, and the member function name starting with **get** is an accessor.

```
(1) const string Car::m_className = "Car";
(2) class Car
(3) {
    (a) public:
        1. Car(int passengers = 5, const string& color = "Red");
        2. string& GetColor ( );
        3. int GetNrPassengers( );
        4. void SetColor (string& color);
        5. void SetNrPassengers (int passengers);
    (4) private:
        1. int m_passengers;
        2. string m_color;
        3. bool m_isMoving;
        4. static string m_className;
(5) };
```

**b) Answer:**

20)(5 points) Given the class defined above, write the constructor using a member initialization list

21)(10 points) Consider the following class definition

```
class IntegerPair
{
public:
IntegerPair (int firstValue, int secondValue);
int getFirst();
int getSecond();
private:
    int m_first;
    int m_second;
};
```

Write the prototype for the prefix and postfix versions of operator++

Write the functions for the prefix and postfix versions of operator++. The function will increment each of the private integer data members.

22)(10 points) Since a constructor cannot return a data type to indicate to the user of a class that invalid parameter values were passed to the constructor, define a mechanism that can indicate to the user of the class if invalid parameters were passed to the constructor.



23)(10 points) Show the output:

```

#include <vector>
#include <iostream>
#include <string>
using namespace std;

const int SIZE = 2;
void print(const vector <int> & aVector);
void print(const vector <bool> & aVector);
void print(const vector <string> & aVector);
void print(const vector <float> & aVector);
int main()
{
    vector <int> a (SIZE);
    vector <bool> b (SIZE);
    vector <string> c (SIZE);
    vector <float> d (SIZE);
    cout << "integer vector" << endl;
    print (a);
    cout << "bool vector" << endl;
    print (b);
    cout << "string vector" << endl;
    print (c);
    cout << "float vector" << endl;
    print (d);
return 0;
}
void print(const vector <int> & aVector)
{
    for (int i = 0; i < SIZE; ++i)
        cout << "vector[" << i << "]: " << aVector[i] << endl;
    cout << endl;
}
void print(const vector <string> & aVector)
{
    for (int i = 0; i < SIZE; ++i)
        cout << "vector[" << i << "]: " << aVector[i] << endl;
    cout << endl;
}
void print(const vector <float> & aVector)
{
    for (int i = 0; i < SIZE; ++i)
        cout << "vector[" << i << "]: " << aVector[i] << endl;
    cout << endl;
}
void print(const vector <bool> & aVector)
{
    for (int i = 0; i < SIZE; ++i)
        cout << "vector[" << i << "]: " << aVector[i] << endl;
    cout << endl;
}
/*
integer vector
vector[0]: 0
vector[1]: 0

bool vector
vector[0]: 0
vector[1]: 0

```

```
string tvector  
vector[0]:  
vector[1]:  
  
float vector  
vector[0]: 0  
vector[1]: 0  
*/
```