CS 1022C-001/002 - Fall 2013 - Midterm Exam
University of Cincinnati

There are 8 questions for a total of 133 points.

Name: _____

Please circle your section : Section 001 - Dr. Talaga : Section 002 - Dr. Helmick

Instructions

- Please read through this entire exam very carefully before starting.
- This exam is closed notes and closed books.
- All work must be written on the exam pages in order to be graded. Any scrap paper used, must be the scrap paper provided during the exam period.
- For programming questions: Please be accurate with your C++ syntax: this includes appropriate use of braces, semicolons, and the proper use of upper/lowercase letters.
- No electronic devices may be used during the exam: this includes (but is not limited to) calculators, phones, tablets, and computers.
- You have 55 minutes to complete the exam.

Good Luck!

Question:	1	2	3	4	5	6	7	8	Total
Points:	36	12	15	15	15	20	20	0	133
Bonus Points:	0	0	0	0	0	0	0	15	15
Score:									

Multiple Choice and True/False

1. Circle the **best** response.

2 (a) Every C++ Program must contain a function called:

A. `getX`
B. `startup`
C. `initialization`
D. `main`

Solution: D - main

2 (b) The input to the compiler is called

A. Source File
B. Machine Code
C. Library Files
D. Executable program

Solution: A - source file

2 (c) The base types `int`, `double`, `long` are unsigned types.

A. True
B. False

Solution: false, they are all signed

2 (d) True of False: It is best practice to initialize variables when you define them.

A. True
B. False

Solution: true

2 (e) What is the preferred way to declare a constant in your C++ program?

A. `#define MONTHS_IN_A_YEAR 12`
B. `const int MONTHS_IN_A_YEAR = 12;`
C. `int MONTHS_IN_A_YEAR = 12;`

Solution: B - `const int MONTHS_IN_A_YEAR = 12;`

2 (f) When calling a function, the default type of parameter passing is:

A. pass-by-reference
B. pass-by-pointer
C. pass-by-value
D. pass-by-osmosis

Solution: C - pass-by-value

2 (g) `if`, `while`, `for` all require the use of curly braces {}:

A. Yes
B. Sometimes they are required

C. No
D. No, but it is good practice to use them

Solution: D - No, but it is good practice to use them

2 (h) True of False: A `for` loop always executes at least one iteration
A. True
B. False

Solution: false

2 (i) True of False: A `while` loop always executes at least one iteration
A. True
B. False

Solution: false

2 (j) True of False: A `do` loop always executes at least one iteration
A. True
B. False
Text

Solution: true

2 (k) True or False: All pointers in your program use the same amount of memory
A. True
B. False

Solution: true

2 (l) When using pointers, the `*` operator is used to
A. access the pointer
B. dereference the pointer
C. assign the pointer
D. inspect the pointer

Solution: B - dereference the pointer

2 (m) If you don't delete all of the dynamically allocated memory in your program, this is called a ...
A. memory leak
B. logical error
C. virus
D. programmer error

Solution: A - memory leak

2 (n) If you attempt to access memory that isn't properly initialized, your program may result in a
A. logical error
B. array index out of bounds exception
C. segmentation fault

D. null pointer error

Solution: C - segmentation fault

2 (o) Accessor functions should be declared with this modifier when possible:

A. `virtual`
B. `accessor`
C. `&`
D. `const`

Solution: D - const

2 (p) True or False: For variables containing an instance of a class, the destructor is automatically called when the variable goes out of scope.

A. True
B. False

Solution: True

2 (q) True or False: For pointers to instances of class types, the destructor is automatically called when the pointer variable goes out of scope.

A. True
B. False

Solution: False

2 (r) A constructor that has a single parameter, that is a reference to an instance of the same class, is called a:

A. Copy constructor
B. Duplication Constructor
C. Assignment operator
D. Destructor

Solution: A - Copy constructor

Short Answer

2. Write a brief response to each question. Please write in complete sentences, using a maximum of 2 sentences.

[6] (a) What is encapsulation? Why is it useful?

Solution: Encapsulation is the process of making data private and internal to a class. This allows you to control how the data is accessed and changed.

[6] (b) How do you discover syntax error? How do you discover logic errors?

Solution: Syntax errors are discovered using the compiler, during program compilation. Logic errors are discovered when running your program or during unit testing.

Code Analysis

3. Examine the following code segment and answer the questions below.

```
1 #include <iostream>
2 using namespace std;
3
4 // todo(helmick): Document this function!
5 int sillyFunction(int x, int &y; const int &z) {
6   x = z;
7   y = 50;
8 }
9
10 int main() {
11   int x = 40;
12   int y = 60;
13   int z;
14   cout << "x: " << x << " y: " << y << " z: " << z << endl;
15   z = y;
16   sillyFunction(z, y, x);
17   cout << "x: " << x << " y: " << y << " z: " << z << endl;
18   return 0;
19 }
```

[5] (a) What is the output of this code?

Solution:
```
x:  40 y:  60 z:  <randomvalue>
x:  40 y:  50 z:  60
```

[5] (b) What is the type of parameter passing used for `x`, `y`, and `z` in `sillyFunction`? Which, if any, parameters will have their changes reflected in the calling code?

Solution: x is pass-by-value, y is pass-by-reference, z is a const reference.
the second parameter, y, can be changed since it is pass-by-reference.

[5] (c) What is missing from `sillyFunction`?

Solution: It should have a return statement, but it doesn't. That is too bad.

4. Examine the following code segment and answer the questions below.

```
1 #include <iostream>
2 using namespace std;
3
4 // todo(helmick): Document this function!
5 int magicTime(int a, int &b; const int &c) {
6   a = c;
7   b = 20;
8 }
9
10 int main() {
11   int a = 10;
12   int b = 30;
13   int c;
14   cout << "a: " << a << " b: " << b << " c: " << c << endl;
15   c = b;
16   magicTime(c, b, a);
17   cout << "a: " << a << " b: " << b << " c: " << c << endl;
18   return 0;
19 }
```

5 (a) What is the output of this code?

Solution:
```
x: 10 y: 30 z: <randomvalue>
x: 10 y: 20 z: 30
```

5 (b) What is the type of parameter passing used for `x`, `y`, and `z` in `magicTime`? Which, if any, parameters will have their changes reflected in the calling code?

Solution: a is pass-by-value, b is pass-by-reference, c is a const reference.
the second parameter, b, can be changed since it is pass-by-reference.

5 (c) What is missing from `magicTime`?

Solution: It should have a return statement, but it doesn't. That is too bad.

5. Examine the following code segment and answer the questions below.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5   int* a = new int;
6   *a = 45;
7   int* b = a;
8   cout << "a=" << *a << " b=" << *b << endl;
9   *b = 55;
10  cout << "a=" << *a << " b=" << *b << endl;
11  b = new int;
12  *b = 65;
13  cout << "a=" << *a << " b=" << *b << endl;
14  *a = 75;
15  cout << "a=" << *a << " b=" << *b << endl;
16  delete a;
17  a = NULL;
18  b = NULL;
19 }
```

[10] (a) What is the output of this code?

Solution:
```
a=45 b=45
a=55 b=55
a=55 b=65
a=75 b=65
```

[5] (b) Does this code correctly clean up all of the memory it dynamically allocates? If not, how would you fix it?

Solution: `b is not cleaned up, it should be deleted instead of just set to NULL.`

Programming Questions

20 6. Design and write the header file for a class that represents a single playing card. Playing cards are immutable, so there don't need to be any methods to change the card.

```
#ifndef CARD_H
#define CARD_H

// Assume that anything you need is included.
using namespace std;

class Card {
```

Solution:

```
public:
  Card(const Suit &suit, int rank);
  Suit getSuit();
  int getRank();
private:
  const Suit suit;
  const int rank;

};
```

20 7. Write a program that reads in one integer from the user and prints a multiplication table from 0 to the number, formatted to so that each number takes up 3 spaces, plus a space between each number.

```
Enter a number: 5
  0   0   0   0   0   0
  0   1   2   3   4   5
  0   2   4   6   8  10
  0   3   6   9  12  15
  0   4   8  12  16  20
  0   5  10  15  20  25
```

```
#include <iostream>
using namespace std;
int main() {
  cout.width(3);
  cout << "Enter the dimension of the multiplication table? ";
  // START ANSWER HERE
```

Solution:

```
  int dim = 0;
  cin >> dim;
  for (int y = 0; y <= dim; y++) {
    for (int x = 0; x <= dim; x++) {
      cout << (y * x);
    }
    cout << endl;
  }
```

```
    // END OF ANSWER
  }
}
```

15 (bonus)

8. Implement a class called `SodaCan` with functions `getSurfaceArea()` and `getVolume()`. In the constructor, supply the height and radius of the can. You may assume that the constant `PI` (type double) is defined and available for you to use. Formulas for your reference:

$$V = \pi r^2 h \textbf{ and } A = 2\pi r^2 + 2\pi rh$$

Solution:

```
class SodeCan {
public:
  SodaCan(double height, double radius) {
    this->height = height;
    this->radius = radius;
  }

  double getSurfaceArea() {
    return 2 * PI * radius*radius + 2 * PI * radius * height;
  }

  double getVolume() {
        return PI * r * r * h;
  }

private:
  double height;
  double radius;
};
```