CSCI 200: Foundational Programming Concepts & Design

Exam 2 Review

1. What is printed?

```
void my_func( int &x, int y ) {
    x = 52;
    y = 7;
}

int main() {
    int x = 0;
    int y = 0;
    my_func( x, y );
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    return 0;
}
```

2. String

Write a function called string_append that receives a string as input and outputs a string.

The function needs to return a string that appends to the parameter the text

" is a super coder."

3. Code

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  int getValue1() const;
  int getValue2() const;
private:
  int _value1;
  int _value2;
};
```

```
// main.cpp
#include <iostream>
using namespace std;

#include "Gnome.h"

int main() {
  Gnome a( 10, 25 );
  cout << a._value1 << " "
       << a._value2 << endl;
  return 0;
}
```

3. Questions

a) What message would the compiler display?

b) Correctly rewrite the line of code to correct the error.

c) What is the purpose of const in the two member functions?

d) What is Gnome() and why doesn't it have a return type?

4. What is legal?

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  int getValue1() const;
  int getValue2() const;
private:
  int _value1;
  int _value2;
};
```

```
// main.cpp - assume appropriate headers
int main() {
  Gnome g1;
  Gnome g2();
  g1._value1 = 52;
  int _value1;
  _value1 = g1.getValue1();
  Gnome g3 = g1;
  g3.g2();
  cout << _value1 << endl;
  cout << _value2 << endl;
  return 0;
}
```

5. Short Answer

- Suppose you have developed a class called MyClass with private data members x and y of type int.
  a) Write the function header for this class's default constructor.
  b) Write the function implementation for this class's default constructor that sets x and y to 0.

6. Functions

```
Circle Circle::doSomething( const Circle &C ) {
  // does something here
}
```

a) What is the name of the function?

b) Is this function a member function? If yes, to what class?

7. Functions cont.

```
Circle Circle::doSomething( const Circle &C ) {
  // does something here
}
```

a) What does the first Circle represent?

b) What does the second Circle represent?

c) What does the third Circle represent?

d) What does the const represent?

8. Constructors

- Which of the following are valid constructors? Justify the issue if one exists.
  a) `BankAccount::BankAccount() const`
  b) `BankAccount::BankAccount( double balance )`
  c) `void BankAccount::BankAccount()`
  d) `BankAccount::BankAccount( const string &acct, double balance )`

9. Member Functions

- Which of the following are valid member functions implementation headers? Justify the issue if one exists.

a) `double HotDog::getPrice() const`

b) `Triangle::calculateArea()`

c) `Buffalo Buffalo::buffalo( Buffalo buffalo )`

d) `void Dog::fetchBall`

e) `double AlarmClock::ring( float )`

10. What is printed?

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  int getValue1() const;
  int getValue2() const;
  int diff();
  int diff( const Gnome &G );
private:
  int _value1;
  int _value2;
};
```

```
int Gnome::diff() {
  return _value2 - _value1;
}
int Gnome::diff( const Gnome &G ) {
  return this->_value2 - G._value1;
}
```

```
int main() {
  Gnome a( 10, 25 ), b( 5, 20 );
  cout << a.diff() << " "
       << a.diff( b ) << endl;
  return 0;
}
```

11. Army of Gnomes!

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, string );
  int getValue1() const;
  string getName() const;
private:
  int _value1;
  string _name;
};
```

- Declare a vector of Gnomes. Then add two Gnomes:
  - harry with value 35
  - sally with value 38

```
int main() {



}
```

12. Composition

```
class Chair { // in Chair.h
public:
  Chair();
  Chair( int, int, int, double );
  // all getters and setters
private:
  int _height, _width, _depth;
  double _price;
};
class Table { // in Table.h
public:
  Table();
  Table( int, int, int, double );
  // all getters and setters
private:
  int _height, _width, _depth;
  double _price;
};
```

- Write a .h file to define a new class DiningSet. DiningSet has two chairs and one table, a bool on whether the set is sold, and a getPrice() function.

13. Composition

a) Write the function implementation of the Chair's default constructor. Use 10.0 for the price and 1 for the height, width, and depth.

b) Write the implementation of getPrice() for your DiningSet class. getPrice() is equal to the sum of the table and chairs price.

14. Pointers

```
01 int a = 5;
02 int b = 6;
03 int *c = &a;
04 int *d = &b;
05 int *e = new int(7);
06 int *f = new int;
07 int *g = new int;
08 f = c;
09 *g = *c;
10 a = 8;
11 *d = 9;
12 *f = 1;
13 *g = 2;
14 *c = 3;
15 delete e;
16 delete f;
17 delete g;
```

1. What is the final value of a & b?
2. What do c, d, e, f, g point to?
3. f is what type of copy of c?
4. g is what type of copy of c?
5. Which of Lines 15, 16, 17 will result in an error? Why? What is the error?

15. Pointers Part 2

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

void foo(int* pX) {
  *pX = 4;
}

void bar(int*& pY) {
  *pY = 5;
}

int main() {
  int *b = new int(2);
  cout << "1 - " << *b << endl;
  foo(b);
  cout << "2 - " << *b << endl;
  bar(b);
  cout << "3 - " << *b << endl;
  return 0;
}
```

1. What is the output?
2. Sketch out the memory usage.

16. Pointers Part 3

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

void foo(int* pX) {
  pX = new int(4);
}

void bar(int*& pY) {
  pY = new int(5);
}

int main() {
  int *b = new int(2);
  cout << "1 - " << *b << endl;
  foo(b);
  cout << "2 - " << *b << endl;
  bar(b);
  cout << "3 - " << *b << endl;
  return 0;
}
```

1. What is the output?
2. Sketch out the memory usage.

17. Analysis

- What is the run time of the following block of code?

```
int matches = 0;
string line1, line2;
getline(cin, line1);
getline(cin, line2);
int shorterLine = min(line1.length(), line2.length());
for(int i = 0; i < shorterLine; i++) {
  if(line1.at(i) == line2.at(i)) {
    matches++;
  }
}
if(line1.length() == line2.length() && line1.length() == matches) {
  cout << "Lines are equal" << endl;
} else {
  cout << "Lines are not equal" << endl;
}
```

18. Analysis 2

- What is the run time of the following block of code?

```
int matches = 0;
string line1, line2;
getline(cin, line1);
getline(cin, line2);
for(int i = 0; i < line1.length(); i++) {
  for(int j = i; j < line2.length(); j++) {
    if(i == j) {
      if(line1.at(i) == line2.at(j)) {
        matches++;
      }
    }
  }
}
if(line1.length() == line2.length() && line1.length() == matches) {
  cout << "Lines are equal" << endl;
} else {
  cout << "Lines are not equal" << endl;
}
```

19. Analysis 3

- What is the run time of the following block of code?

```
int matches = 0;
string line1, line2;
getline(cin, line1);
getline(cin, line2);
for(int i = 0; i < line1.length(); i++) {
  for(int j = i; j < line2.length(); j++) {
    if(i == j) {
      if(line1.at(i) == line2.at(j)) {
        matches++;
      }
      break;
    }
  }
}
if(line1.length() == line2.length() && line1.length() == matches) {
  cout << "Lines are equal" << endl;
} else {
  cout << "Lines are not equal" << endl;
}
```

20. Analysis 4

- Of Questions 17, 18, 19:
  - Which have the best performance?
  - The worst?

21. The Big 3

- What are the Big 3?
- What is the Rule of 3?
- Why should we follow the Rule of 3? What can occur if we don't?
- How do the Big 3 relate to shallow/deep copies? What is the difference between the two?

22. Programming Paradigms

- What is the difference between Procedural Programming and Object-Oriented Programming?

- Write an example block of code that illustrates each style in use.

23. File I/O

- Given a file named “xc.txt” with the following data

  n

  x_1 x_2 x_3 … x_n

  - Where the first integer in the file, n, states how many integers will follow in the file (n will be at least 1)

- Write a program to read in all the integers and print out the largest & smallest integer.