CSCI 10: Introduction to Computer Science
Fall 2016 — Michael J. Bannister

# Final Exam

December 9, 2016

**all problems are of equal value**

Name (please print): _____

Student ID: _____

**Problem 1:** For this problem read the following program and answer the questions below.

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

void rotate(int a, int b, int c);
void print(int a, int b, int c);

int main() {
    int i = 1, j = 2, k = 3;
    for (int c = 0; c < 3; c++) {
        print(i, j, k);
        rotate(i, j, k);
    }
    return 0;
}

void print(int a, int b, int c) {
    cout << "(" << a << "," << b << "," << c << ") ";
}
void rotate(int a, int b, int c) {
    int d = a;
    a = b;
    b = c;
    c = d;
}
```

**Question 1:** The program is meant to print (1,2,3) (2,3,1) (3,1,2). What does it actually print?

**Question 2:** How would you fix the program to get the expected output?

**Problem 2:** In this problem, you need to implement a function `bool happy(string s);` which returns `true` if the string contains the two "smiley" characters :) and `false` otherwise. The function should only return `true` if the characters : and ) are next to each other and in the correct order.

```
bool happy(string s) {




}
```

**Problem 3:** In this problem, you need to implement `string space_score(string s)` a function that replaces whitespace characters with underscores. More specifically the returned string is the same as the input string `s` except all whitespace characters have been replaced with underscores `_`. To determine if a character is a whitespace use the `cctype` library function `bool isspace(char c);` which returns `true` if `c` is a whitespace character and `false` otherwise.

```
string space_score(string s) {




}
```

**Problem 4:** In C++, functions cannot return more than one value. In class we discussed several ways to work around this limitation. In this problem, you will discuss two of these techniques.

**Technique 1:** Explain how call-by-reference can be used to effectively return more than one value from a function. You may use an example if that helps your explanation.

**Technique 2:** Explain how returning a `struct` can be used to effectively return more than one value from a function. You may use an example if that helps your explanation.

**Problem 5:** In this problem you will write a program to read in the lines of a file `input.txt` and if the length of a line is less than 80, write the line to the file `output.txt`. Some of the boring code has already been provided for you.

```cpp
#include <fstream>
#include <string>

using namespace std;

int main() {
    ifstream in_file;
    in_file.open("input.txt");
    ofstream out_file;
    out_file.open("output.txt");



    in_file.close();
    out_file.close();

    return 0;
}
```

**Problem 6:** For this problem, read the following program and determine its output (what is printed). Please box your answer to separate it from your scratch work.

```cpp
#include <iostream>

using namespace std;

void swap(int &a, int &b);
int green(int &a);
int blue(int b);
void print(int a, int b);

int main() {
    int a = 2, b = 7;
    print(b, a);
    b = blue(a);
    print(b, a);
    a = green(b);
    print(b, a);
    swap(b, a);
    print(a, b);
    print(a++, ++b);
    return 0;
}

void swap(int &a, int &b) {
    int temp = b;
    b = a;
    a = temp;
}

int green(int &a) {
    a *= 2;
    return a + 1;
}

int blue(int b) {
    b += 1;
    return b * 2;
}

void print(int a, int b) {
    cout << "(" << a << "," << b << ")" << endl;
}
```

**Problem 7:** If we list all the natural numbers below 10 that are multiples of 3 or 5, we get the list 3, 5, 6 and 9. The sum of these multiples is 23. Write a program to find the sum of all the multiples of 3 or 5 below 10000 and print the answer. Some of the boring code is given to you below.

```cpp
#include <iostream>

using namespace std;

int main() {




    return 0;
}
```

**Problem 8:** Write a function to determine if two arrays of `int` values are equal. Arrays are considered equal if they have the exact same values in the exact same order. You are free to name and design your function as you like, but you must provide a short description describing how you expect it to be used, i.e., describe how you expect the input parameters to be used.