Practice Problems for the SI204 Final

1 Programs and Function Definitions

1. Data for marathon finishing results is kept in a file. For each runner you get the last name, the time, and the finishing position (i.e. 1 for first place, 2 for second place, etc. The entries are in alphabetical order by name. Read in the results file and print out the first five finishers names and times, in order from first to 5th. Below is an example input file and the proper program output:

| Adams 2:38:22 4 |
| Brown 3:01:00 8 |
| Croft 2:22:18 1 |
| Davis 3:08:13 9 |
| Jones 2:48:53 5 |
| Kraft 2:58:58 6 |
| Moots 3:12:34 10 |
| Ricks 3:00:58 7 |
| Tweed 2:35:43 2 |
| Zanie 2:36:32 3 |

Top five finishers are:
Croft 2:22:18
Tweed 2:35:43
Zanie 2:36:32
Adams 2:38:22
Jones 2:48:53

Create a class Runner for this, and note that no sorting is necessary to do this ... make clever use of an array of five Runner objects!

2. Define a class Pos (stand for “position”) and any necessary functions such that the following program, implementing a particularly cruel version of the where’s waldo game, works:

Version for CDR Vincent’s Sections

```cpp
int main()
{
    // Randomly place waldo
    Pos waldo(rand() % 9, rand() % 9);

    // Guess until waldo’s found!
    Pos guess;
    int count = 0;
    do {
        cout << "Guess position: ";
        guess.read();
        count++;
    } while (!waldo.equal(guess));

    // Output # of guesses needed
    cout << "You found waldo in ";
    << count << " guesses!" << endl;

    return 0;
}
```

Version for Other Sections

```cpp
int main()
{
    // Randomly place waldo
    Pos waldo(rand() % 9, rand() % 9);

    // Guess until waldo’s found!
    Pos guess;
    int count = 0;
    do {
        cout << "Guess position: ";
        read(guess);
        count++;
    } while (!equal(waldo, guess));

    // Output # of guesses needed
    cout << "You found waldo in "
        << count << " guesses!" << endl;

    return 0;
}
```
3. Write a code fragment that declares a 2-dimensional array of integers called \( \text{array2D} \), with both dimensions of size 50, and initialize the elements along the diagonal (i.e. \( \text{array2D}[0][0], \text{array2D}[1][1], \text{array2D}[2][2], \text{etc} \)) to 1 and the rest of the elements to 0.

4. Given the following definition of the class \text{Customer}:

   class \text{Customer} 
   { 
     public: 
       int id; 
       string name; 
   }

   write a function \text{int find(string s, Customer *A, int n);} that takes a string \( s \) and an array \( A \) of \( n \) \text{Customer} objects, and returns the index of the element of \( A \) whose \text{name} is \( s \). If no such element exists, return \( n \).

2 Short Answer

1. Answer each of the following with true or false:

   (a) A function may return an object of user-defined type, i.e. of a class.
   (b) The types of the parameters received by a function may be classes.
   (c) The default constructor must have the same name as the class.
   (d) All constructors must have the same name, but different arguments.
   (e) Two member functions of two different classes can have the same name.

2. Find the errors in the following:

   class \text{Submarine} 
   { 
     public: 
       string captain = "Ramage"; 
       int depth = 500; 
     
     \text{Submarine}() 
     { 
       cout << "Submarine Created w/ default values" << endl; 
     } 
     \text{Submarine}(string name, int setdepth) 
     { 
       cout << "Submarine Created w/ input values" << endl; 
       captain = name; 
       depth = setdepth; 
     } 
   };
3. Find the errors in the following:

```cpp
class Submarine
{
public:
    string captain;
    int depth;
    int Submarine()
    {
        cout << "Sub created!" << endl;
        captain = "Fluckey";
        depth = 200;
        return depth;
    }
    int depthfathoms()
    {
        return depth/6;
    }
};
```

4. For the following:
Suppose you have the following class definitions and function prototypes:

```cpp
class Student
{
public:
    string name;
    double *grade;
};
bool operator==(Student stu, string str);
void swap(int &r, int &s);
```

And suppose you have the following variables defined:

```cpp
int i, j, k, **M;
double x, y, *Z;
Student S;
char a, b, *A;
string v, w;
ifstream IN("data.txt");
```

Explain why each statement won’t compile:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Reason for compile-time error</th>
</tr>
</thead>
<tbody>
<tr>
<td>swap(x, k)</td>
<td></td>
</tr>
<tr>
<td>S.length()</td>
<td></td>
</tr>
<tr>
<td>A[i][j]</td>
<td></td>
</tr>
<tr>
<td>Z[S]</td>
<td></td>
</tr>
<tr>
<td>S + x</td>
<td></td>
</tr>
<tr>
<td>swap(i+j, k)</td>
<td></td>
</tr>
<tr>
<td>S == w</td>
<td></td>
</tr>
<tr>
<td>S == w == v</td>
<td></td>
</tr>
</tbody>
</table>