Introduction to C++: COMP 322 (Winter 2014) - Quiz #1

Name: ID:

Instructions

This is a *closed book* quiz. There are seven questions on 4 pages, for a total of 20 points.

In the code fragments, assume that the context (e.g. appropriate header files and `using namespace` statements) has been specified correctly.

When asked what a code fragment would print, don't worry too much about whitespaces and newlines. I care about the output numbers, and not about formatting.

Problem 1 (3 pts)

What would the following piece of code print?

```
int data[] = { 1 , 20, 300, 4000};

int* z = data + 3;
int k = (z - data) - 1;
int& r = *(data + k);
r = 5;
*z = 60;

cout << data[0] << " " << data[1] << " ";
cout << data[2] << " " << data[3] << endl;
```

Problem 2 (3 pts)

I would like `main()` to print `11 11`. How should I call the functions `foo1` and `foo2`?

```
void foo1(int* x) {
  *x += 10;
}

void foo2(int& x) {
  x -= 10;
}

int main() {
  int a = 1, b = 21;
  foo1(   );
  foo2(   );
  cout << a << " " << b;
}
```

Problem 3 (3 pts)

Implement a function `rotate` that “rotates” an input array. To understand what would happen to an array if we rotate it, check the main function below. It should print `3 1 2 40 10 20 30`.

```
/* rotate: takes input an array z of size size_z */
void rotate(int* z, int size_z);

int main() {
  int x[] = {1, 2, 3};
  int y[] = {10, 20, 30, 40};
  rotate(x, 3);
  rotate(y, 4);
  cout << x[0] << " " << x[1] << " " << x[2] << " ";
  cout << y[0] << " " << y[1] << " ";
  cout << y[2] << " " << y[3];
}
```

Problem 4 (3 pts)

The following compiles, but it is not an example of proper memory management. Please explain why and fix the code accordingly.

```
struct StudentGrade {
  string name;
  int grade;
};

studentGrade* enterGrade(string ss, int gg) {
  StudentGrade toReturn;
  toReturn.name = ss;
  toReturn.grade = gg;
  return &toReturn;
}
```

Problem 5 (3 pts)

The following does not compile. Please allocate memory for a 2D array of size 20 by 20 using proper memory management techniques.

```
double** allocate2Darray() {
  double** toReturn = new double[20][20];
  return toReturn;
}
```

Problem 6 (3 pts)

Implement two functions that work with the same struct StudentGrade. The first function should take input a vector of such structs and return a list containing the same elements. The second function should take as input a list of such structs and return a map that uses the name of a struct as key, and the associated grade as the value in the map. Make sure the map object that you return contains all the associations that are included in the input list.

```
struct StudentGrade {
  string name;
  int grade;
};

list<StudentGrade> vecToList(vector<StudentGrade>& vec);
map<string, grade> structToMap(list<StudentGrade>& lst);
```

Continue solution to Problem 6:

Problem 7 (2 pts)

I would like to write some piece of code that first prints to cout the value of an integer x, and then it replaces the value of x using data coming from cin. Explain why the following does not compile, and rewrite the code to do the right thing.

```
int x = 3;
cin >> (cout << x);
```

Use the extra white space as you wish: