Program Design Midterm Exam

1. Pointers and Dynamic Arrays

(a) Explain the concept of a pointer in C++.

(b) What are the outputs produced by the following codes?

```
double *p1, *p2; p1 = new double; p2 = new double;
*p1 = 1.5; *p2 = 3.6; cout << "p1, p2 = " << *p1 << ", " << *p2 << endl;
p2 = p1; cout << "p1, p2 = " << *p1 << ", " << *p2 << endl;
*p2 = 5.6; cout << "p1, p2 = " << *p1 << ", " << *p2 << endl;
*p1 = -0.1; *p2 = 6.2; cout << "p1, p2 = " << *p1 << ", " << *p2 << endl;
double *p3;
p3 = new double; p3 = p1; *p3 = 7.3; cout << "p1, p2, p3 = " << *p1 << ", " << *p2 << ", " << *p3 << endl;
```

(c) Consider the following code:

```
char *A; A = new char [50];
```

Write code to fill the array A with 50 characters typed in from the keyboard.

(d) What is the output of the following code fragment?

```
int ArraySize = 20;
int *p; p = new int [ArraySize];
int *a = p; int i; for (i = 0; i < ArraySize; i++) p[i] = -1 * i; a[19] = 19;
for (i = 1; i < ArraySize; i += 2) cout << p[i] << " "; cout << endl;
```

2. What is the output of the following C++ codes?

```
char a[5] = {'x', 'y', 'z', 'p', 'q'};
for (i = 2; i < 5; i++) cout << a[i] << " ";
a[1] = a[4]; for (i = 0; i < 4; i++); cout << a[i] << " ";
```

3. Suppose we expect the elements of the array A to be ordered so that A[0] ≤ A[1] ≤ A[2] ≤ …
However, to be safe we want our program to test the array and issue a warning in case it turns out that some elements are out of order. The following C++ code is supposed to output such a warning, but it contains a bug. What is it? (5%) Please also correct this code. (5%)

```
double A[30]; <Some code to fill the array A goes here>
for (int index = 0; index < 30; index++)
    if (A[index] > A[index + 1])
        cout << "Array elements " << index << " and " << (index + 1) << " are out of order.";
```

4. Is the following program fragment legal? (2%) If so, what is the output? (8%)

```
int main()
{ vector <int> a(10); int i;
  vector <int> b;
  for(i = 0; i < a.size(); i++) a[i] = i * i;
```

```
b = a; a[1] = -1;
for(i = 0; i < b.size(); i++) cout << b[i] << " ";
return 0; }
```

5. Consider the following code (and assume that it is embedded in a complete and correct program and then run it):

```
string s1, s2;
cin >> s1 >> s2; cout << s1 << " x " << s2 << "->END";
```

If the input from the keyboard is shown as follows, what's the output?

A String is a Joy Forever !

6. Which of the following declarations are equivalent?
   (a) *char* StringVar[10] = "HELLO";
   (b) *char* StringVar[10] = {'H', 'E', 'L', 'L', 'O', '\0'};
   (c) *char* StringVar[10] = {'H', 'E', 'L', 'L', 'O'};
   (d) *char* StringVar[6] = "HELLO";
   (e) *char* StringVar[ ] = "HELLO";

7. Consider the following two code fragments (and assume that it is embedded in a complete and correct program and then run it):
   (a) *char* MyStr[80];
   cin.getline(MyStr, 10); cout << MyStr << "->END";
   If the input from the keyboard is shown as follows, what's the output?
   May the hair on your toes grow long and curly.
   (b) *string* s;
   getline(cin, s); cout << s << "->END";
   If the input from the keyboard is shown as follows, what's the output?
   A String is a Joy Forever !

8. Given the following class definition, write an appropriate definition for the member function set.

```
class Temperature
{
public:
   void set(double new_degrees, char new_scale);
   // Sets the member variables to the values given as arguments.
   double degrees;
   char scale;    // 'F' for Fahrenheit (華氏溫度) or 'C' for Celsius (攝氏溫度).
};
```

9. Write a definition of a structure type for records consisting of a person's wage rate (denoted by *double* wage_rate), vacation (number of days, denoted by *int* vacation), and status (hourly (H) or salaried (S), denoted by *char* status). Denote this structure type as EmployeeRecord.

10. Given the following structure definition. What will be the output produced by the following code?

```
#include <iostream>
using namespace std;
struct ShoeType
{
        char style;
        double price;
};
int main()
{
      ShoeType shoe1, shoe2;
      shoe1.style = 'C';
      shoe1.price = 8.88;
      shoe2 = shoe1;
      shoe2.price /= 4;
      cout << shoe1.style << " $" << shoe1.price <<endl;
      cout << shoe2.style << " $" << shoe2.price <<endl;

      return 0;
}
```