

PIC 10A Final Exam Answers**Fall 2006****Prof. Todd Wittman**

1.) [20 points] For each library listed below, give a brief description of the library and its use. Then give an example of a function in the library and what it does. The first one is done for you as an example.

<cmath> Contains useful mathematical functions like exponential and trig functions.
 sqrt(x) -- returns the square root of a double x

<iostream> Handles input / output for the console window.
 cout << x -- outputs the value of x to the console window

<vector> Allows declaration of vector class (dynamic list) and its member functions.
 v.push_back(x) – adds element with value x to the end of the vector v

<string> Allows use of the string class, which is used for parsing text.
 s.length() -- returns the # characters in string s

<iomanip> Formats output: control spacing and number of digits reported.
 setw(w) -- outputs data right-justified in column of width w

<fstream> Used for input / output from data files.
 ofstream o – creates an output file stream o for writing to a file

2.) [10 points] Briefly comment each of the following lines of code. The first line is done for you. Make sure you describe the difference between the last two lines.

```
int x = 2;           //Creates integer x and assigns it value 2.
int *p;             //Creates a pointer p to a integer.
p = &x;             //Sets p pointing to the address of variable x.
*p = 4;             //Changes the value that p points to equal to 4.
cout << p;          //Prints out the address that p points to.
cout << *p;         //Prints out the value in the block p points to (4).
```

3.) [15 points] Each of the code segments below has a compile error. Correct the compile errors. You will get points off for changing code that was not an error.

a.) int hobbit = 3; if 0 < hobbit < 5 ++hobbit; if (0 < hobbit && hobbit < 5)	b.) vector<string> vec[3]; vec[0] = "Hello Frodo.\n"; vec[1] = "Hi Gandalf."; vec[2] = vec[0] + vec[1]; Use parentheses not brackets in first line: vec(3)
c.) double *p; *p = 42.1; Dangling pointer error. Illegal because p does not point to a block, so the value 42.1 has nowhere to go: double *p = new double;	d.) int a[10]; for (int i = 1; i <= 10; i++) a[i] = i*i; Arrays start at 0: for (int i=0; i < 10; i++)
e.) double investment = 2000.0; double return = 1.016 * investment; cout << "The return on your investment of \$" << investment << " is \$" << return; return is a reserved word, change the name of the variable to myReturn (or something)	

4.) [20 points] Write the output of the following segments of code.

a.) int a = 4; int b = 3; bool isEqual; do { if (b > 2) b = a%b; else a--; isEqual = (a==b); cout << "a=" << a << " b=" << b << "\n"; } while (!isEqual);	a=4 b=1 a=3 b=1 a=2 b=1 a=1 b=1
---	---

b.)	<pre>int x = 8; int y = 3; cout << y+x*y << x / y << y / x;</pre>	2720
c.)	<pre>int good = 15; cout << good << "as\new" << good%4;</pre>	15as ew3
d.)	<pre>int a[5]; a[4] = 3; for (int i=3; i >= 0; i--) { a[i] = 2*a[i+1] - i; cout << a[i]; }</pre>	34714

5.) [20 points] Write the output of the following program. Track your variables in a walkthrough table as we did in class.

<pre>#include <iostream> using namespace std; void fun1 (int *p, int *q) { *p = 100; p = p + 2; *p = *q; *q = *(p+1); cout << "p=" << *p << " q=" << *q << "\n"; } bool fun2 (int x[], int size) { int *p = x; int *q; q = p; for (int i=1; i<size; i++) { x[i] += x[i-1]; cout << x[i] << " "; } cout << "\n"; fun1 (p, q); return (*p == *q); } int main () { int a[5] = {2,4,6,8,10}; if (fun2 (a,5)) cout << "true\n"; else cout << "false\n"; for (int i=0; i<5; i++) cout << a[i] << " "; return 0; }</pre>	<pre>6 12 20 30 p=100 q=20 true 20 6 100 20 30</pre>
--	--

6.) [15 points] Write a function `vectorAverage` that returns the average of the elements of an integer vector. The prototype is written below.

```
double vectorAverage (vector<int> v) {
    int sum = 0;
    for (int i=0; i < v.size( ); i++)
        sum += v[i];
    return (double) sum / v.size( );
}
```

7.) [20 points] Write a function `insertElement` that inserts a given integer `x` into an integer vector before a given position `pos`. The prototype is written below.

(This problem is in Lecture 26 Notes.)

```

void insertElement (vector<int> &v, int pos, int x) {
    v.push_back (v[v.size( )-1]);
    for (int i = v.size( )-1; i > pos; i--)
        v[i] = v[i+1];
    v[pos] = x;
    return;
}

```

8.) [25 points] Write a full program, starting from #include, that gets the name of a text file from the user. The program then counts the number of words and lines in the file. You may assume the user's file exists and contains only words, no numbers or isolated punctuation marks. You may also assume that all line breaks occur immediately after a word and there are no blank lines. A sample run is shown below for the input file at right.

Enter file name: input.txt	input.txt
# words = 6	Hello Gandalf.
# lines = 3	How are you?
	Good?

```

#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main ( ) {
    string fileName;
    cout << "Enter file name: ";
    cin >> fileName;
    ifstream fin;
    fin.open (fileName.c_str( ));
    string word;
    int numWords = 0;
    int numLines = 1; // #lines = #line breaks + 1
    char c;
    while (fin >> word) {
        numWords++;
        fin.get(c);
        if (c == '\n')
            numLines++;
    }
    cout << "# words = " << numWords << "\n# lines = " << numLines;
    return 0;
}

```

9.) [20 points] Write a function replaceVowels that takes a string as input and replaces every lower-case vowel (aeiou) with an one of the four symbols below:

* & # @

The symbol used should be selected at random. A sample run is shown below.

```
cout << replaceVowels ("Frodo, look out for the orcs!!!");
```

Output: Fr&d*, l#*k *@t f&r th# &rcs!!!

```

string replaceVowels (string s) {
    int r; //Random number.
    for (int i=0; i < s.length( ); i++) {
        if ( s[i] == 'a' || s[i] == 'e' || s[i] == 'i'
            || s[i] == 'o' || s[i] == 'u') {
            r = 1 + rand( ) % 4;
            if (r==1)
                s[i] = '*';
            else if (r==2)

```

```

        s[i] = '&';
    else if (r==3)
        s[i] = '#';
    else
        s[i] = '@';
    }
}
return s;
}

```

10.) [35 points] Write a class Car that keeps track of the name of the car, the fuel efficiency in miles per gallon (mpg), the number of miles driven, and the number of gallons left in the gas tank. The constructor should take the car name and the fuel efficiency as input. Write the class declaration first and then define the member functions for the class. You'll have to figure out what functions to add based on the sample usage below.

```

Car my_car ( "Ferrari Testarossa", 20 ); //My Ferrari gets 20 miles to the gallon.
my_car.add_gas (10); //Add 10 gallons of gas to the tank.
my_car.drive (100); //Drive 100 miles.
cout << my_car.get_name( ) << " has driven " << my_car.get_miles( ) << " miles and has "
    << my_car.get_gas( ) << " gallons left in the tank.";

```

Output: Ferrari Testarossa has driven 100 miles and has 5 gallons left in the tank.

```

class Car {
public:
    Car (string new_name, double new_mpg);
    void add_gas (double amount);
    void drive (double dist);
    string get_name ( );
    double get_miles ( );
    double get_gas ( );
private:
    string name;
    double mpg;
    double miles;
    double gas;
};

```

```

Car :: Car (string new_name, double new_mpg) {
    name = new_name;
    mpg = new_mpg;
    miles = 0;
    gas = 0;
}

```

```

void Car :: add_gas (double amount) {
    gas += amount;
    return;
}

```

```

void Car :: drive (double dist)
{
    miles += dist;
    gas -= dist/mpg;
    return;
}

```

```

string Car :: get_name ( ) {
    return name;
}

```

```

double Car :: get_miles ( ) {
    return miles;
}

```

```

double Car :: get_gas ( ) {
    return gas;
}

```