

Exam 2

CSI 201: Computer Science 1
Fall 2017

Professor: Shaun Ramsey, Ph.D.

I understand that this exam is closed books and closed notes and is to be completed without a calculator, phone, or other computer. I am **NOT** allowed to use any external resources to complete this exam. The work that I am submitting and that I have viewed during this exam is mine. I understand that images, video and sound may be taken and recorded during this exam. I have completed this exam in accordance with the Washington College Honor Code.

For full credit, remember to use good style and programming practice throughout.

Name: _____

Signature: _____

1. 20 points Write the output that each code snippet would produce. Leave it blank if there is no output. Each line of output is worth 2 points. Assume that the code snippets are continuous. They are separated here to correlate output with your answers.

Code Snippet:	Console Output:
<pre>//fill up the vector vector<int> filler; for (int i = 0; i < 11; i++) { filler.push_back(i + 3); } //output some of the entries //and other information cout << filler.at(0) << endl; cout << filler.at(2) << endl; cout << filler.size() << endl;</pre>	
<pre>//output some entries //using a strange start for (int i = 3; i < 7; i++) { cout << filler.at(i + 1) << endl; }</pre>	
<pre>//moving some values //walk through this carefully filler.at(2) = filler.at(3); filler.at(3) = filler.at(1); filler.at(1) = filler.at(2); cout << filler.at(1) << endl; cout << filler.at(2) << endl; cout << filler.at(3) << endl;</pre>	

2. Concepts: Answer the following briefly. When code is requested, your response should consist of less than 2-3 lines of code.

(a) 6 points What does fun1 return in the following function calls:

```
fun1(33.0);
```

```
fun1(0.0);
```

```
fun1(-33.0);
```

```
//f1 is given by this function
bool fun1(double funny) {
    if (funny != 0) { //truly funny
        return true;
    }
    else { //not so funny
        return false;
    }
}
```

(b) 2 points Name one situation in which we use call by reference and explain why.

(c) 2 points What happens when an invalid index is accessed in a vector?

- (d) 6 points What does squareMax return when the following are called?

```
squareMax(4, 3);
```

```
squareMax(3,11);
```

```
squareMax(3, 9);
```

```
double squareMax(double var, double limits) {  
    double value = var * var;  
    if (value > limits) {  
        value = limits;  
    }  
    return value;  
}
```

- (e) 4 points Examine the function squareMax from above. If we pass 9 as the second parameter, for what values of the first parameter will the function return 9? This is extremely useful, for example, if we wanted to write a unit test in a for loop to test a variety of values for the first parameter. In that situation, we might be able to assert that the function returns 9 in these situations. Give the full range of possible values for full credit. List a handful of values for partial credit.
- (f) 2 points Show a function call of the function squareMax above that returns a value of 13.

(g) 4 points Demonstrate how to read in (from the console) a vector of 3000 strings named **names**.

(h) 4 points Demonstrate how to output (to the console) every other element of a vector of doubles named **grades**.

(i) 2 points Describe the run-time error with the following code snippet.

```
vector<int> pumpkins(42);  
for(unsigned i = 1; i <= pumpkins.size(); i++) {  
    pumpkins.at(i) = 42 - i;  
}
```

3. 10 points In this problem, you have been given a vector of integers that represent ages and is aptly named **ages**. You are required to determine how many elements of the vector fall between the values of 17 and 22 (inclusive). Output (to the console) this amount as well as the total number of elements that are in the vector.

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

int main() {
    // assume the vector of integers named ages is defined here and
    // has been filled with all the data required
    ...
    // Your code goes here

    // end of your code
    return 0;
}
```

4. 10 points Compound interest helps us understand how much money can be earned (or lost) by saving (or taking a loan). To compute how much money we will have (or owe) after a certain number of years (t) at a given rate (r) when we put in (or take out) a certain amount of dollars (P), uses the equation: $A = P(1 + r)^t$. We've been asked to write a function in which P , r and t are given to the function. In return, the function computes and returns A . The `cmath` library provides the `pow(x,y)` function. `pow(x,y)` can be used to compute x^y .

```
#include <iostream>
#include <assert>
#include <cmath>
using namespace std;

double futureValue(double P, double r, double t);

int main() { //some unit tests
    assert( futureValue(1000, 0.0, 10) == 1000);
    assert( futureValue(1000, 0.1, 1) == 1100);
    assert( futureValue(1000, 1.0, 3) == 8000);
    assert( futureValue(1000, .01, 10) == 1104.52 );
    return 0;
}

double futureValue(double P, double r, double t) {
    \\your code begins here
```

```
\\your code ends here
}
```

Question	Points	Score
1	20	
2	32	
3	10	
4	10	
Total:	72	