Final Exam

CSI 201: Computer Science I
Fall 2017

Professors: Shaun Ramsey

Question	Points	Score
1	20	
2	26	
3	12	
4	20	
Total:	78	

I understand that I should not discuss this exam with anyone before Saturday. 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. README - Answer in 1-3 lines of code or 1-2 sentences where appropriate. In general, you are meant to assume that any described variables are used throughout a program before your code snippet will run (with the exception of the first question). You are meant to assume your code is running in main.

(a) [2 points] Show how to declare a variable to hold the precise height (with several decimal places) of a dog. Name this variable `my_height`.

```
double my_height;
```

(b) [2 points] Output to the console "`hearts`" when an integer variable named `card_pos` lies between the values 13 and 25 (inclusive).

```
if(card_pos >= 13 && card_pos <= 25) {
  cout << "hearts" << endl;
}
```

(c) [2 points] Assuming your program is properly seeded, show how to get a random value between 1 and 6. Store the result in a variable named `die`.

```
die = rand()%6 + 1
```

(d) [2 points] Write a loop that increments (increases by 1) the variable, `apples_index`, 5 million times. Note: for credit you must do this with a loop.

```
for(int apples_index = 0; apples_index < 5000000; apples_index++){

}
```

(e) [2 points] Consider the following class definition:

```
class Animal {
  public:
      string name;
      string type_of_mammal;
};
```

Given two `Animal` variables (with variable names `Spot` and `Socks`), give code that outputs "SAME" if the two variables have the same `type_of_mammal` values.

```
if(Spot.type_of_mammal == Socks.type_of_mammal) {
 cout << "SAME" << endl;
}
```

(f) [2 points] Given a pointer to an Animal named `animal_ptr`, show how to set that object's `name` value to ``Fuzzy''.

```
animal_ptr->name = "Fuzzy";
```

(g) [2 points] Show how to open a file for input. The file is named `dogs.txt`.

```
ifstream fin("dogs.txt");
```

(h) [2 points] What does a compiler do?

translates human readable source code (we write in C++) into machine readable executable code -- it's a translator!

(i) [2 points] Does the compiler run (or execute) your program? Circle yes / no. You may give a brief explanation of your answer.

NO! The compiler translates it. We must then run that executable.

(j) [2 points] What is a pointer?

An address. The value of a pointer is the address of another piece of memory.

2. Code Output.

(a) [20 points] Determine what is printed by each code snippet.

Code Snippet:	**Output:**
`int sum = 3;` `cout << sum / 2 << endl;`	1

`int card_rank = 21;`
`if(card_rank%13 < 9) {`
`  cout << card_rank%13 + 2 << endl;`
`}`
`else {`
`  cout <<  "special" << endl;`
`}`
	10

`int i = 7;`
`while (i <= 10) {`
`    cout << i << endl;`
`    i = i + 3;`
`}`
`cout << i << endl;`
	7
	10
	13

`vector<int> place_id(500);`
`for (int i=0; i < place_id.size(); ++i) {`
`    place_id.at(i) = 2 * i + 5;`
`}`
`cout << place_id.at(2) << endl;`
	9

`cout << sqrt(9) + 1 << endl;`	4

`int a = 3;`
`int b = 5;`
`int c = 7;`
`a = b;`
`b = c;`
`c = a;`
`cout << c << endl;`
	5

`cout << 10%2 << endl;`	5

`for(int i = 0; i < 2; ++i) {`
`  cout << i + 2 << endl;`
`}`
	2
	3

(b) [4 points] Consider the following function. Notice that it is call by reference, so modifications affect the originally passed object.

```
void halved(vector<int> &incoming) {
   for(int i = 0; i < incoming.size(); ++i) {
      incoming.at(i) = incoming.at(i) / 2;
   }
}
```

What is the output when we run the following code snippet?

```
vector<int> mine;
for(int i = 0; i < 2000; ++i) {
   mine.push_back(i);
}
cout << mine.at(10) << endl;
halved(mine);
cout << mine.at(10) << endl;
halved(mine);
halved(mine);
cout << mine.at(10) << endl;
```

10
5
1

(c) [2 points] Consider this function definition:

```
int f(int a, int b) {
   if(a == 0)  //curly brackets removed around if/else for brevity
      return b;
   else
      return 2 * a;
}
```

What is the output when we run the following code snippet?

```
int a = 2;
int b = 0;
cout << f(a,b) << endl;
cout << f(b,a) << endl;
```

4
2

3. [12 points] Debugging. The following function is meant to compute the force between two masses using Newton's law of universal gravitation. But, it has some errors. Find all four bugs in this code. For each bug, write down a description of what is wrong. Then write what the mistake was and lastly write on what line it occurs.

```
1.  double gravity_force(double mass_one, double mass_two, double distance) {
2.     double constant = 0.00000000006674;
3.     double force = 0.0
4.     if(mass_one > 0 && mass_two > 0 & distance > 0) {
5.        force = constant * mass_one * mass_2 / (distance*distance);
6.     }
7.     else {
8.        cerr << "Error in parameters to gravity_force function;
9.     }
10.    return force;
11. }
```

Error 1. Missing Semi-Colon Line #: 3

Error 2. Missing & should be && Line #: 4

Error 3. mass_2 is undeclared, should be mass_two Line #: 5

Error 4. Missing ending `` for string literal Line #: 8

4. [20 points] Programming Choose 2 of a,b and c! Each one is worth 10 points.

   (a) In this code you will be writing a function to determine if one number is evenly divisible by another. We do this by checking the modulus(%) of one number with another. When the modulus is 0, then the first number is evenly divisibly by the second. In your function, pass in these two numbers. If the first (`num1`) is evenly divisible by the second (`num2`), then this function should evaluate to true. Otherwise the function should evaluate to false. In main, write some code that uses your new function. Get user input for two integers and output "evenly divisible" if the first is evenly divisible by the second. Be sure to use your function in this main code.

```
bool isDivisible(int num1, int num2) {
   if(num1%num2 == 0) { //evenly divisible
      return true;
   }
   else {
      return false;
   }
}
int main() {
   int a,b;
   cin >> a >> b;
   if(isDivisible(a,b)) {
      cout << ``evenly divisible'' << endl;
   }
}
```

(b) You've been given a `vector` of many strings (you don't know exactly how many as you write the code). The vector is named `crypto`. This vector is actually a secret message, but it must be decoded first. Write some code that walks through every string of crypto. If the string is "apples" then the next word should be output to the console. If the word is "paprika", then no more words should be output to the console. Show a code snippet that takes this vector of strings named `crypto` and outputs the secret message. An example vector of strings might be "I", "did", "love", "apples", "run", "over", "with", "a","dash", "of", "paprika", "but", "I", "hate", "apples", "now". Following the rules above, only "run" would be output to the console.

```
//assume: a vector of strings named crypto already filled
//an alternate method uses the variable nomore
bool nomore = false;
for(int i = 0; i < crypto.size(); ++i) {
  if(crypto.at(i) == "apples" && i < crypto.size() - 1) {
    if(nomore == false) { //only prints if paprika not seen
      cout << crypto.at(i+1) << endl;
    }
  }
  else if(crypto.at(i) == "paprika") {
    break; //break will quit the loop and thus no more prints
    //alternatively use the following line
    nomore == true;
  }
}
```

(c) You've been asked to write the virtual component of a conveyor belt of cucumbers. Cucumbers come from two different lines, let's call them `left` and `right`. These cucumbers are in jars but they may not be full. 10 cucumbers fit into a single jar. In the assembly line, one jar of cucumbers is simply dumped into another. Any spillage is caught underneath to serve a different need. The process continues after that point with simply the one (potentially filled) line of jars of cucumbers. You will be given two vectors, `left` and `right`. Make a new vector that is the element by element addition of `left` and `right`. However, the new value may not go above 10. If the vectors are of different lengths, assume that the jars continue untouched. For example, if `left` is 3,7,5,4 and `right` is 5,5,5, then you should create a vector that is 8, 10, 10, 4. This comes because the first jar on the left is 3 and the first jar on the right is 5. 3+5 is 8. The second jars are 7 and 5 and 7+5 = 12. *Isn't that a pickle?* Since 12 is more than 10, 2 cucumbers spill out of the jar, but the jar continues full. The third jars are 5 and 5. 5+5=10 which is okay. Lastly, there is no fourth jar on the right, but the left jar will continue on with 4. So after the conveyor belts merge, the number of cucumbers continuing on the assembly line is: 8, 10, 10, 4. This code requires no interaction with the console. Assume the `left` and `right` vectors are filled and managed elsewhere. You don't know their sizes as you write this code.

```
//assume left and right exist
//they're filled elsewhere and have some unknown amts
int min_size = left.size();
int max_size = right.size();
if(left.size() > right.size()) {
   min_size = right.size();
   max_size = left.size();
}
vector<int> newjars;
for(int i = 0; i < min_size; ++i) {
   newjars.push_back(left.at(i) + right.at(i));
   if(newjars.at(i) > 10) {
     newjars.at(i) = 10;
   }
}
for(int j = min_size; j < max_size; ++j) {
   if(j < left.size()) {
     newjars.push_back(left.at(j));
   }
   else{ //could add if j < right.size() as a sanity check
     newjars.push_back(right.at(j));
   }
}
```