

Name: _____

Section: _____

1. Provide the correct Unix command for each of the following. (1 point each)

<code>nedit george.cc</code>	edit source code named <code>george.cc</code>
<code>g++ -o seeGeorgeRun george.cc</code>	compile <code>george.cc</code> to an executable <code>seeGeorgeRun</code>
<code>cd cs150x</code>	change the current directory to your class directory
<code>ls -l</code>	list all files in the current directory, including size
<code>submit -d test1.cc</code>	submit a program named <code>test1.cc</code>

2. For each of the following possible identifier names, indicate (by circling) if it is valid or invalid. If it is invalid, state why. (2 points each)

<code>nona</code>	valid / invalid	VALID _____
<code>first@name</code>	valid / invalid	INVALID: special characters (@) not allowed
<code>2ndName</code>	valid / invalid	INVALID: must begin with a letter

3. List three (3) of the C++ data types used in CMPS 150.
(1 point each)

<code>int</code>	<code>char</code>
<code>double/float</code>	<code>bool</code>
<code>string</code>	

4. Use the declarations and initializations listed below to evaluate the following arithmetic expressions, based on the C++ language. (total 10 points)

```
int    a = 5, b = 8, c = 13, d = 64;  
double e = 12, f = 6.5, g = 3.875, h = 18.4;
```

<code>f + d / a * e - h</code>	<code>6.5 + 64 / 5 * 12 - 18.4</code>	132.1
	<code>6.5 + 12 * 12 - 18.4</code>	
	<code>6.5 + 144 - 18.4</code>	

<code>d - h + f / a</code>	<code>64 - 18.4 + 6.5 / 5</code>	46.9
	<code>64 - 18.4 + 1.3</code>	
	<code>45.6 + 1.3</code>	

<code>c % b - g * 2</code>	<code>13 % 8 - 3.875 * 2</code>	-2.75
	<code>5 - 7.75</code>	

5. Using the declarations listed below, write a valid C++ statement that will divide the variable **george** by the variable **bertha** and store the result in **answer**. Be sure to write the code such that if there are any decimals in the quotient, they are retained when stored in answer. (4 points)

```
int    george, bertha;
double answer;                answer = double(george) / bertha;
```

6. Write two (2) different lines of code, both of which will create a blank line in the output of a program. (1 point each)

```
cout << endl;

cout << "\n";
```

7. Write the code to ask the user for a number (integer) that indicates the number of seconds it took to complete a race, and get their input. Then write the code to calculate and output that time in hours, minutes and seconds. For example, if the user enters **9630**, the program should output **2:40:30**
NOTE: Do not worry about declaring/using constants, just write the input/process/output statements. (2 points for prompt/input, 3 points for calculation, 5 points for output)

```
cout << "Enter seconds: ";
cin >> seconds;
```

```
hours = seconds / 3600;
```

```
minutes = (seconds - hours * 3600) / 60;
```

```
remainMins = seconds - hours * 3600 - minutes * 60;
```

```
cout << hours << ":" << minutes << ":" << remainMins << endl;
```

8. Write the code to ask the user for their hometown (which may contain spaces), and get their input. (2 points)

```
cout << "Enter your hometown: ";
getline(cin, home);
```

9. Using the following input statements and user input, indicate what will be stored in each variable. The variable name is indicative of the data type it is able to store. (4 points each input line)

```
cin >> int1 >> double1 >> char1 >> string1;
```

Enter data: 45.6ABC DEF GHI

```
int1          45
double1       .6
char1         A
string1       BC
```

```
cin >> string1 >> double1 >> char1;
```

Enter data: 45.6ABC 123 GHI

```
int1          "junk"
double1       123
char1         G
string1       45.6ABC
```

10. Use the declarations and initializations listed below to evaluate the following logical expressions, based on the C++ language. (total 10 points)

```
int    a = 5, b = 9, c = 13, d = 64;
double e = 14, f = 6.5, g = 3.875, h = 18.4;
```

```
f > d || a * e == h && g <= c          FALSE
F   ||      F   &&   T
F   ||      F
```

```
d % a == c - b          TRUE
4   ==   4
```

```
!(g / h)                FALSE
!(3.875/18.4)
!(T)
```