SCIT-EIS-UOW
CSCI251/CSCI851 Advanced Programming
Autumn 2019

Laboratory Exercise 1 (Week 2)

1 Part One: Variables and a little bit of input

1. Make sure you know how to declare and set literal values for each of the following basic types:

   ```
   int, double, float, char, string
   ```

2. In the preliminary exercises you were supposed to have prepared a source file `Hello.cpp`. Modify the code so you can request a name from the user, read in the name, and say Hello name, instead of hello world, where name is the name specified by the user. It would make sense to store the read in name in a string.

3. If you run the last program and enter a number as the name, what happens?

4. Modify the last program so instead of storing the name in a string you store it in a float. What happens when you enter you name now?

2 Part Two: Error messages

Most weeks I'll provide some buggy code, and get you to learn from experience about the messages the compiler gives you and hopefully improve your ability to read code. Here go a few code fragments that will get you started. I suggest trying these with both `CC` and `g++`, to see the different form of feedback. Sometimes one compiler is more informative than another.

1. A complete program, use just these lines, no headers...

   ```
   int main()
   {
   cout << "Hello" << endl;
   }
   ```

2. With appropriate headers ...

   ```
   int value = 5;
   cout >> value >> endl;
   ```

3. With appropriate headers ...

   ```
   int value;
   cout << "Enter an integer : ";
   cin << value;
   ```

4. With appropriate headers, some of these should work. Determine which don't work and explain what is happening in each case.

```
int value = 2;
cout << value << 2 << endl;
cout << (value << 2) << endl;
cout << value >> 2 >> endl;
cout << (value >> 2) >> endl;
```

3 Part Three: More I/O, functions, a loop

1. Write a program in a file `t3a.cpp` to read an integer from the user and output 10 times that value.

2. Write a program in a file `t3b.cpp` to read an integer N from the user and output the N times the values 1 to 12. Output these values in a sensible format. You should write a loop to do this. The basic loop syntax will be covered in the lecture next week but we have seen an example in the lectures (**page 20 of S2b**).

```
for( int counter=0; counter < limit; counter++ )
{
     ...
}
```

3. Rewrite `t3b.cpp` as `t3c.cpp` with the calculation and printing in a function.

4. Rewrite `t3c.cpp` as `t3d.cpp` with the calculation and printing in distinct functions. At this point don't do all the calculations first and then the printing, calculate and print each multiplication entry one by one. You shouldn't have a collection of values stored at the end, only N.

4 Part Four: Multiple files and functions

Write two C++ programs to determine the cost of building a table with a square top.

- Version One: Everything in the same file.

- Version Two: Spread the program across multiple files.

The `main()` function should call the following seven functions. You should, if you need to give information to the function, pass variables so the functions make copies of them and accept return values. The seven functions are:

1. A function which accepts from `standard in` the number of chairs to go with the table.

2. A function which accepts from `standard in` the surface area (in m^2) of the table.

3. A function which accepts from `standard in` any colour for the cushions on the chair seats. Any input should be accepted as a colour and recorded. Use a string.

4. A function which accepts from `standard in` the type of wood used to build the table and chairs; 'm' for mahogony, 'o' for oak or 'p' for pine. Ideally any other entry should be rejected. To deal with this input look at using an `if` statement, or switch.

5. A function that takes the number of chairs, the surface area of the table and the type of wood and calculates the price. The price for a table with surface area S and N chairs is $x(S + \frac{1}{2}N)$ where x is \$200, \$150 and \$95 respectively for mahogony, oak and pine.

6. A function to display the details of the purchase, including the dimensions of the table.

7. A function to display the price of the purchase.