# Lab 2 - Variables

**Lab Topics**

1. Using a variable.
2. Using int vs float variables.
3. Working with char variables.
4. Exploring types of errors.

## Directions

- The labs are marked based on attendance and effort.
- It is your responsibility to ensure the TA records your progress by the end of the lab.
- Do each step of the lab in order.
- While completing these labs, you are encouraged to help your classmates and receive as much help as you like. Assignments, however, are individual work.
  **You may not work on assignments during your lab time.**
- If you complete the lab early, you should experiment with C++; however, you may leave if you prefer.
- If you do not finish the lab exercises during your lab time, you are encouraged to complete them later to finish learning the material. You will still receive full marks if you arrived on-time and put in your best effort to complete the lab.

## 1. Setup

Refer to Lab 1 for more information on how to complete this section.

1. Create a new VC++ program named `lab2`.
2. To start with, make it output a `"Hello"` message to the screen, and prompt the user to "Enter any character to exit the program." at the end.

## 2. Experiment with a Variable

1. Remove your hello message from above and change it to display a header like:
   ```
   SONG LIBRARY
   ************
   ```
2. Inside the `main()` function, declare a variable named `songs`, of type `int`, and initialize it to `0`:
   ```
   int songs = 0;
   ```
3. Output the value of `songs` to the screen:
   ```
   cout << "Initial number of songs: "<< songs << endl;
   ```
   Your output should look like:
   ```
   Initial number of songs: 0
   ```

4. Add a statement which assigns a new value of 10 to songs:

   ```
   songs = 10;
   ```

5. Add a statement which outputs the variable to the screen. Your added output should look like:

   ```
   Number of songs after first purchase: 10
   ```

6. Finally change the value of `songs` to 654 and again output to the screen. The combined output should look like:

   ```
   SONG LIBRARY
   ************
   Initial number of songs: 0
   Number of songs after first purchase: 10
   Number of songs after today: 654
   ```

7. **Understanding:**
   What is the difference between defining a variable and using a variable?
   How can you tell where a variable is defined?
   What happens if you try to redeclare a variable?
   Test this by changing your `songs = 10;` statement to `int songs = 10;`

   **Suggestion:** Type your answers to these "Understanding" questions inside your lab's program as comments. This way you'll be able to refer back to your answers later if the TA asks, or when studying for a test.

## 3. int vs float

1. Add a new heading output which outputs:

   ```
   CALCULATE COSTS:
   ****************
   ```

2. Create **two** `float` variables.
   Name the first one `purchase` and initialize its value to 16.83.
   Name the second one `tax` and set it to 0.12.

   Your declaration for `purchase` might look like:

   ```
   float purchase = 16.83;
   // <Your declaration & initialization of tax goes here!>
   ```

3. Create a new variable of type `float` named `totalAsFloat` and initialize it to the cost of the item with tax:

   ```
   float totalAsFloat = purchase * (1 + tax);
   ```

4. Create a new variable of type `int` and name it `totalAsInt` and initialize it to the cost of the item with tax:

   ```
   int totalAsInt = purchase * (1 + tax);
   ```

5. Write two `cout` statements which show the values in each of your total variables. Have each `cout` statement displays the cost of the item without, and with tax.

```
Cost after tax of 16.83 as a float: 18.8496
Cost after tax of 16.83 as an int: 18
```

**Note that your output statement must not hard-code the numbers: it must use the values stored in the variables.** Any time an example is given (such as here with the cost 16.83), you cannot hard-code your program to use just that number: you must complete what is being asked, which in this case is to perform a calculation using variables, one of which happens to be initialized to 16.83. For example, your `cout` statement **<u>cannot</u>** be:

```
cout << "Cost after tax of 16.83 as a float: 18.8496" << endl;
```

6. Edit the statement where the variable `purchase` is initialized. Change its value to 1000. Predict the output and rerun the program and verify your prediction. Note the value may be off by 1 from what you expect; we will look at this later in the course.
7. **Understanding:**
   Based on your investigation in this section, describe the difference between using a variable of type `int` vs of type `float?` List two situations for each type where you would use that type of variable.

## 4. Exploring char

1. Add a new heading-output to your `main()` function to show:

   ```
   ASCII TESTING:
   **************
   ```

2. Add the following lines of code to your program (copy and paste will save you time). You must correct the formatting (tabs) once you have added the lines:

   ```
   char c1 = 67, c2 = 77, c3 = 80, c4 = 84;
   char n1 = 49, n2 = 50, n3 = 53;
   cout << "Magic code = " << c1 << c2 << c3 << c4;
   cout << " " << n1 << n2 << n3 << endl;
   ```

3. Based on the following ASCII table, predict what the above code will output (BEFORE YOU RUN THE CODE! ;) ). For example, to read the table you might look up capital H (6th row) and see its associated ASCII code is 72.

   ```
   ASCII Table:
   ************
    =32     !=33    "=34    #=35    $=36    %=37    &=38    '=39
   (=40     )=41    *=42    +=43    ,=44    -=45    .=46    /=47
   0=48     1=49    2=50    3=51    4=52    5=53    6=54    7=55
   8=56     9=57    :=58    ;=59    <=60    ==61    >=62    ?=63
   @=64     A=65    B=66    C=67    D=68    E=69    F=70    G=71
   H=72     I=73    J=74    K=75    L=76    M=77    N=78    O=79
   P=80     Q=81    R=82    S=83    T=84    U=85    V=86    W=87
   X=88     Y=89    Z=90    [=91    \=92    ]=93    ^=94    _=95
   `=96     a=97    b=98    c=99    d=100   e=101   f=102   g=103
   h=104    i=105   j=106   k=107   l=108   m=109   n=110   o=111
   p=112    q=113   r=114   s=115   t=116   u=117   v=118   w=119
   x=120    y=121   z=122   {=123   |=124   }=125   ~=126
   ```

Note that a space (i.e., pressing the space bar) is ASCII code 32.

4. Run the code and check your prediction.
5. Add another couple of lines of code to output your first name (or first and last name if you have a very short first name). Use a similar technique for displaying your name one character at a time by setting variables of type char to the correct ASCII value.
6. **Understanding:**
   What is the relationship between characters and numbers within the computer?

## 5. Errors

1. Recall from lecture that there are three types of errors: compile-time, run-time, and logical errors. If you do not recall the types, refer to the notes before continuing.
2. For each type of error, how would you expect to the error to manifest (or become visible) while writing a program? In other words, how would you know you had a compile time error, a run-time error, or a logic error?
3. Analyze the following code for errors without running it through the compiler. There is one error of each type. Identify the errors and consider how you would correct the errors.

```
cout << endl << "ERROR TESTING:"<<endl;
cout         << "**************"<<endl;
int a = 10, b = 3, c = 0;

c = (a * b) / 2;
cout << "Average of "<<a<<" and "<<b<<" is: "<<c<<endl;

c = a + b
cout << "Sum of "<<a<<" and "<<b<<" is: "<<c<<endl;

b = 0;
c = a / b;
cout << a<<" split zero ways is "<<c<<endl;
```

4. Copy the above code into your program and verify your predictions.
5. **Understanding:**
   How would you recognize each type of error in a program you are writing?
   Since each type of error manifests itself differently, how could you use this to identify problems in your code?

## 6. Extra Challenge

If you have extra time, try these tasks for an extra challenge. (Not for credit).

- Add to your program a section which calculates the area of a square and a cube based on a given length.
  - What happens when you use a huge length, such as 2000000000?
  - Try storing the result as a long and a double. Which one works with a huge length?

## 7. Skills and Understanding

You should now be able to answer all the "understanding" questions in the previous sections. Complete the following to get credit for the lab:

- Show the TA the following:
  - Your operational program which completes all of the above outputs, including showing your name.
  - The TA may ask you to explain any section of the lab, or answer any of the "Understanding" questions.
- **Nothing** is to be submitted electronically or in hard-copy for this lab.