Lab 3 - Input and Expressions

Lab Topics

1. Working with order of operations and casting (implicit and explicit).
2. Reading data from the keyboard.
3. Writing simple programs which perform calculations.

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. cin Troubles

1. On the next page is some code to a program which you would like to generates the following output; however, there are currently some bugs in the code. You must understand what's going wrong, and then fix the errors so that your program generates the same output as below.
   (Remember, you cannot hardcode the output, you must use variables as appropriate).

```
Enter the name, and form, or the commodity: Gold bars
Enter how many kg of Gold bars you have: 0.0009
Enter the value (per kg) of Gold bars: 48191.83

Your 0.0009 kg of Gold bars is worth $43.3726!

Enter any letter to continue.
```

2. Copy and paste the following code into a file called `commodity.cpp,` and then debug the application so it generates the output shown on the previous page.

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

int main () {
    // Enter the name of the commodity
    cout << "Enter the name, and form, or the commodity: ";
    string name;
    cin >> name;

    // Enter the amount:
    cout << "Enter how many kg of " << name << " you have: ";
    int amount_kg = 0;
    cin >> amount_kg;

    // Enter the price:
    cout << "Enter the value (per kg) of " << name << ": ";
    int valuePerKg = 0;
    cin >> valuePerKg;

    // Calculate total value and output to the screen
    int totalValue = amount_kg + valuePerKg;
    cout << endl;
    cout << "Your " << amount_kg << " kg of " << name
          << " is worth $" << totalValue << "!" << endl;

    // Hold the window open:
    char holdOpen = 0;
    cout << endl << "Enter any letter to continue.";
    cin >> holdOpen;

    return 0;
}
```

3. **Understanding:**
    1. What did you have to do to get the name and form of the commodity to be entered into the computer?
    2. What did you have to do to have the quantity and prices be correct?
    3. Based on the discussion in class about buffered input, how does this exercise show that C++ uses buffered inputs?

2. Age in Days

- Write a program named `daysOld.cpp` which, using a single `cin` statement, reads from the user his/her first name and age (in years). Then, write out to the screen a message of the form

  ```
  "Hello ____! On your last birthday you had lived ___ days!"
  ```

  - Ignore leap years.
  - Hint: there are 365 days per year.

- Modify your program to also the display the number of decades (10's of years) which the user has lived. Your program should display the fractional part of the answer, if any.

- Sample interaction (may look different if you choose):

  ```
  Please enter your first name, and your age (in years) separated by
  a space.
  Mickey 83
  Hello Mickey! On your last birthday you had lived 30295 days!
  This is equivalent to 8.3 decades.
  ```

- Comment your program. You must:
  - Comment at the top of the file with your name, ID, and student number.
  - Comment at the top of the file which briefly describes the program.
  - Comment every section of your code. You should have at least one comment per 3-4 lines of code, often more.

3. Cookies!

- A certain bag of cookies claims to contain 50 cookies. It states that there are 10 servings in the bag, and each serving is 365 calories.

- Write a program called `cookies.cpp` which asks the user how many of these cookies they ate, and report the total number of calories they consumed.
  - Make each of the hard-coded values (such as number of cookies per bag) a named constant. Be sure to capitalize the names based on our naming conventions.
  - Sample interaction (may look different if you choose):

    ```
    Please enter the number of cookies you ate: 2
    The 2 equates to 146 calories.
    ```

  - If you have problems getting it correct, make sure each of your constants is correct by printing them to the screen. Try working through the math by hand and seeing where things go wrong in your program.

- Add reasonable comments to your code, as described above.

4. Temperature Conversion

- Write a program which converts temperatures from Celsius to Fahrenheit.
  - The conversion formula (where F is degrees Fahrenheit, and C is degrees Celsius) is:

    $$F=\frac{9}{5}C+32$$

- Your program should prompt the user for the current temperature and then convert it. Display the final result including the decimal portion of the temperature.
  - Sample interaction (may look different if you choose):

    ```
    Please enter the temperature in degrees C: 20.1
    20.1'C equates to 68.18'F.
    ```

- Add comments to your code.

- **Understanding:**
  - How can you read in an integer number vs a floating point number?
  - How do order of operations and implicit/explicit type casting work in your program?

5. Extra Challenge

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

- Modify your temperature converter to also convert the temperature into degrees Kelvin [K]:

  $$K=C+273.15$$

- Modify your temperature converter to also convert the temperature into degrees Rankine [R]:

  $$R=\frac{9}{5}K$$

6. 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 programs which complete all of the above tasks.
  - 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.