Lab 4 - If & Loop

Directions

• You are encouraged to help your classmates, and to receive as much help as you like. Assignments, however, are individual work.
  You may not work on assignments during your lab time.

1. Number Checking

Create a new C++ program named ifcheck.cpp which does tasks listed below.

Notes:
- Since this program uses arbitrary values, don't worry about using named constants to start.
- Sample output is on the next page.

1. Prompts the user to enter their favourite number (as an integer).
2. If the number is equal to 42, then using two print statements, print “Awesome choice!” on one line, and “That’s mine too!” on a second line.
3. Independent of previous checks (i.e., as its own separate if/else/else.. statement), if the number is:
   - negative: write “How negative of you!”
   - zero: write “That’s nothing!”
   - positive: write “How positive of you!”
4. Independent of previous checks, if the number is:
   - even: write “Now you are just getting even with me.”
   - odd: write “How odd!”
   Hint: Check for odd using the mod (%) operation.
5. Independent of previous checks, if the number is:
   - less than 10, write “Small.”,
   - between 10 and 50 inclusive, write “Medium.”,
   - greater than 50 and less than 100, write “Large.”,
   - between 100 and 1000 inclusive, write “Huge.”,
   - greater than 1000, write “Massive!”
   HINT: Use an if-else if... structure to check non-overlapping regions (like above).
   Such as (different numbers, but similar idea):
   ```cpp
   if (x < 5) {
   } else if (x <= 30) {
   } else if (x < 80) {
   } else {
   }
   ```
6. Independent of previous checks, if the number is greater than 10,000 then:
   1. Print the number of thousands in the number. For example, if given the number 21456, print “That’s 21k!”.
      Hint: what math operator can you apply that will extract just the thousands?
2. Using a nested if statement (inside your check for greater than 10,000), if the number is:
   - evenly divisible by 1000, print “With nothing left over.”
   - otherwise, print the number left over once the thousands are stripped away. For example, given the number 21456, print “With 456 left over.”

7. Sample Outputs:
   42⁴:
   
   Enter your favourite number: 42
   Awesome choice!
   That's mine too!
   How positive of you!
   Now you are just getting even with me.
   Medium.

   Negative:
   Enter your favourite number: -5
   How negative of you
   How odd!
   Small.

   Massively odd:
   Enter your favourite number: 12345
   How positive of you!
   How odd!
   Massive!
   That's 12k!
   With 345 left over.

   Zero:
   Enter your favourite number: 0
   That's nothing!
   Now you are just getting even with me.
   Small.

   Larger than 10,000:
   Enter your favourite number: 21456
   How positive of you!
   Now you are just getting even with me.
   Massive!
   That's 21k!
   With 456 left over.

   Enter your favourite number: 951000
   How positive of you!
   Now you are just getting even with me.
   Massive!
   That's 951k!
   With nothing left over.

1 If only we knew the question...
2. **Going Loopy**

A while loop allows you to execute one or more statements multiple times. For example

```cpp
int i = 0;
while (i < 4) {
    cout << "Value: " << i << endl;
    i++;
}
```

1. Create a new program inside your project named `loopy.cpp`
2. Have your program ask the user for a number \( N \) (an integer), and then print all numbers from \( N \) down-to 0.
3. Modify the program to only print out even numbers from \( N \) down-to 0 inclusive. There are many ways you can make it do this; any way is fine.
4. Independent of the above work, sum all numbers from 0 up to \( N \). Print the answer.
   - *Hint: You’re using the same \( N \) the user entered above. Your previous loop may have wanted to change the value, so you should have created a new variable to store a copy of the user’s value for later use.*
   - *Hint: You’ll need a new variable to hold the sum of values.*
5. Independent of the above work, multiply all numbers from 1 up to \( N \). Print the answer. This is the value of \( N \)-factorial.
   - *Hint: You’ll need a new variable to hold the product of numbers. You’ll want to start this variable at 1, so when you later multiply it by the numbers between 1 and \( N \) it correctly computes the result.*
6. Sample outputs:
   
   Enter N: **2**
   All even numbers 2..0 = 2, 0,
   Sum 0..2 = 3
   Product 1..2 = 2

   Enter N: **5**
   All even numbers 5..0 = 4, 2, 0,
   Sum 0..5 = 15
   Product 1..5 = 120

   Enter N: **20**
   All even numbers 20..0 = 20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0,
   Sum 0..20 = 210
   Product 1..20 = -2102132736

7. **Understanding**

   Explain why \( N = 20 \) gives an unexpected value.
3. Extra Challenge Tasks
These are not required to complete the lab, but interesting exercises to try if you have time:

1. Change all the magic numbers in ifcheck.cpp to be named constants. 
   Hint: Use refactor → extract constant.

2. Check if your code matches the suggested coding style guide (see assignments section of course website for link).

3. Review a friend's implementation of both parts of the lab and find bugs. Here are some things to look for:
   1. Correct indentation.
   2. Correct logic on test cases.
   3. Boundary cases on their checks.
   4. Inconsistent use of { … } 

Nothing is to be handed in for this lab if completing the lab during lab time.