while and do while

https://csci-1301.github.io/about#authors

June 15, 2021 (12:05:32 AM)

Contents

In Part 1 of this lab you practice implementing while loops with different terminating conditions and applied to different types of problems.

In Part 2 of this lab you will practice using do while loops.

1 Practicing while Loops

Write a loop that displays...

1. all integers between 0 and 100 that are divisible by 3.
2. all integers between 1 and 100 but do not display numbers between 10-20 and 40-60.
3. `Hello World` ten times.
4. 10 rows of text where `Hello` is displayed on every other row and `World` on every other row
5. value that decreases by 0.5 on each iteration. Start from 10.0 and iterate as long as the value remains positive.

In these problems you will need to ask for user for input then perform described actions on the input.

1. Ask the user for a positive integer value (1 or greater). After the user provides a number, compute the result of this series where n represents the number obtained from user: $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + ... + \frac{1}{n}$

2. Assume you have the following mapping of zip codes to cities:

zip code	city name
30901	Augusta
30904	Augusta
30912	Augusta
30809	Evans
30813	Grovetown
30814	Harlem
30815	Hephzibah
29842	North Augusta

Write a small program that asks the user to enter a zip code. Then display to the user the city name

based on the zip code user entered. If the zip code is not included in the table given above, the program should display "unknown city!". Allow the user to repeat the input until the user indicates they are ready to quit by entering approprite sentinel value.

2 Do while Loops

Before writing code, think through the following problems:

1. In your own words, what is the difference between `while` and `do while` loops?
2. Can you think of a problem where you can use `while` but not `do while` loop?

Next solve the following problems using a do while loop:

1. Display numbers 0 to 50.

2. Display numbers 30 to -20.

3. Write a do while loop that generates this output:

   ```
   1 10 100 1000 10000 100000 1000000
   ```

4. Ask the user to enter positive integers. Keep track of the largest number user enters. After user indicates they are done, display the largest number user entered.