Metropolitan State University

ICS 140 Computational Thinking with Programming

Class Exercise 6

**Lecture Section**

1. What is the function of a repetition structure?

Repeatedly execute the same piece of code while some condition is met or not(met)

1. What is a condition-controlled loop?

A loop that will continue to repeat itself until the condition is true

1. What is an infinite loop?

A loop that continuously repeats itself infinitely, usually when the conidition will never be false and break the loop

1. What is a term that represents a variable that keeps a running total in a loop?

Accumulator

1. What is an Augmented Assignment Operator?

+=, -=, /=

1. What is a sentinel?

Value in a loop that causes termination of the loop

1. How does an input validation loop work?

The loop will only execute after it has checked the input and made sure that it is in the range or values desired. It will rerun until it is validated.

**Reading while statements**

Write what the code prints

1. Code:

i = 1

while i <= 5:

    print(i)

    i += 1

Prints: 1, 2, 3, 4, 5

1. Code:

i = 5

while i >= 1:

    print(i)

    i = i - 1

Prints: 5, 4, 3, 2, 1

1. Code:

i = 1

j = 7

while i < j:

    print(i,j)

    i += 2

    j -= 2

print("End",i,j)

Prints:

1,7

3,5

End 5 3

**Writing while loops**

Write the python code for the following situations. I have highlighted variable names in bold.

1. Start with **total** set to 0. Write a while loop that prompts the user to enter a dollar amount for each item they purchase. Add the cost of each item to **total.** Repeat the process until **total** is 100 or more. Print the **total** cost at the end.

Total = 0

While total <100

Total += float(input(“enter dollar amount: “))

Print(total)

1. Start with **total** set to 0. Write a while loop that prompts the user to enter a dollar amount for each item they purchase just like the previous problem and add the cost of each item to the **total**. Repeat the process until the user enters 0 for the cost to indicate they are done. Print the **total** cost at the end.

Dollar = -1

Total = 0

While dollar != 0

Dollar = float(input(“enter dollar amount: “))

Total += dollar

Print(“Your final total is: “)

1. Create a while loop for input validation. Ask the user to enter a number between 1 and 10. If they enter a number outside of this range, repeat the loop and prompt them again. If they enter a number between 1 and 10, exit the loop and print the number.

Number = int(input(“Please enter a number between 1 and 10”))

While number < 1 or number > 10

Print(“Error. Please enter a valid input”)  
 Number = int(input(“Please enter a number between 1 and 10”))

Print(number)

**Programming Exercise**

For the following exercise, you will write a simple game. The purpose will be to have the user guess a number between 1 and 10.

* The program will use the random library to generate a random number each time.
* If the user guesses high or too low, the program should tell them so.
* The program should keep track of how many guesses the user has made
* When the user guesses correctly, the program should congratulate them and tell them how many guesses they made.

It should look something like this when run:

Text

Description automatically generated

Copy the python code in the section below.

**Python Code**

*import* random

guessAttempts = 1

number = random.randint(1,10)

print("I am thinking of a number between 1 and 10")

print("Pssst! its",number) *# Keep in temporarily for testing porpoises*

userGuess = int(input("Please enter a guess 1 through 10 : "))

*while* userGuess != number:

*if* userGuess > number:

print("Too high! Please guess a lower number!")

*else*:

print("Too low! Please guess a higher number!")

guessAttempts += 1

userGuess = int(input("Please enter a guess 1 through 10 : "))

print("Congratulations! You guessed correctly!")

print("It only took you",guessAttempts,"attempts!")

Take a screenshot of an example run of the program and paste it below.

**Example Output**

**Text

Description automatically generated**