Computer Science Principles

Class 4

Today's Topics

- Review of "if –else" and "functions"
- Introduce Case Statement decision block
- Introduce Loops
- Github Work – again
- Class Project Work

If – else if - else

- Any Questions?
- Homework examples?
- View if-else document on github

Bad Idea – Long “if – else if -- else

- If(a == 1)
- print (a was number one)
- Else if (a == 2)
- print (a was number two)
- Else if (a == 3)
- print (a was number three)
- .
- .
- .
- .
- Else if(a == 100)
- print(a was number 100

Better Idea – Case Statement

- Variable x;
- Case x
- when 0 then print “x was zero”
- when 1 then print “x was one”
- when 2 then print “x was two”
- Else
- print “x was something else”
- End

Multiple items

- Case x
- when 1
- print “the value is one”
- x += 1
- print “now the value is two”
- when 2
- print “The value is two”
- when 3 then print “the value is three”
- end

Getting Loopy

- Almost all programs will involve a series of repeated steps.
- To save time typing repeated steps, its better to form a code loop.
- Usually loops are used to repeat for a certain number of times, or until a certain condition is met.

While Loops

- While (condition is TRUE)
- steps to repeat
- steps to repeat
- End

- Notice indenting like the “if – else” blocks

Dangers of while loops

- Ever had a program or app “freeze” or lock up?
- It was probably stuck in a loop!
- Bad coding can result in an unanticipated condition that causes the loop to remain ”TRUE”

Infinite Loop

- While(TRUE)
- This is repeated forever
- End

- While (1) #remember a number != zero is true
- repeated steps
- end

While loops should escape

- Variable x = 5
- While (x < 10)
- x += 1
- End

- Will loop, adding 1 to variable x, and once x == 10, will exit the loop

Special: Do - while

- Do
- steps to repeat
- steps to repeat
- While (test is TRUE)

- What's different?

Do – While vs While

- While loops have the test at the first line, so its possible that no loop statements execute
- Do – While allows the loop statements to execute at least once.

For loops, primarily counting

- For (x = 0; x < 10; x+= 1)
- repeated steps
- End

- Format: “x = 0” is initializer
- “X < 10” is test for escape
- “x += 1” ending action of every loop
- For(initializer, test, action) general terms

Count to 1 million

- For(x=0; x<1000000; x += 1)
- End

No repeated steps, but loop will execute. Why?

This loop is essentially a “delay” loop.

For loops can be broken apart into a while loop

For loop -> While loop

- For(x=0; x < 100; x += 1)
- End

- Is the same as this while loop
- X = 0
- While(x < 100)
- x += 1
- End

Make two cups of coffee

- For(x = 0; x < 2, x+= 1)
- make cup of coffee
- end

Good practice in loops: Constants

- Constant NUM_CUPS = 2

- For(x = 0; x< NUM_CUPS; x+= 1)
- make cup of coffee
- End

Count by 2’s

- For (x = 0; x < 100; x += 2)
- print x
- End

- 0,2,4,6,8,…… 96,98 #why not 100 printed?

Homework

- Write what is printed by the code on the next page.
- Variables:
- while_count = 0
- NUM_LOOPS = 2
- X, Y

```
While (while_count < NUM_LOOPS)
    For (x= 0; x<NUM_LOOPS; x+=1)
        For(y=0;y<NUM_LOOPS; y+=1)
            print x, y
        end                     #end of y for loop
    end                         #end of x for loop
    while_count += 1
End                             #end of while loop
```

Additional Homework

- Read over “Operators” in Class4 folder.
- Review “Data Types” file in Class4 folder, and answer quiz sheet at end of it. Bring to next class.