CPSC 231: Review

Review of some points before your actual midterm

James Tam

Know Your Syntax

- Reminder: Look at my notes when I specify the 'format' for the syntax of a new construct or concept.

- **Format:**

```
def <function name>(<parameter 1>,<parameter 2>...
  <parameter n-1>, <parameter n>):
```

- **Example:**

```
def display(celsius,fahrenheit):
```

- One example, naming conventions: which of the following is proper name for a variable?

```
age
2ndName
First name
_first
```

James Tam

- Which of the following is proper name for a variable?

```
age
2ndName
first name
_first
```

- Only the first one
- Last option violates style requirements

James Tam

Data Types

- Just because something looks similar means that the type of the data is the same.
- Determine the output of the following:
  - Example 1: `print("45" + "20")` Strings: concatenation
  - Example 2: `print("12"/2)` Strings: cannot divide
  - Example 3:
    - `var = input()`
    - `var = var / 8`
  - Example 4: one would never (or at least should never) write code like this python will treat this as valid expression.
    - `if(True == "True"):`

Python allows

bool string

Q: is a bool a string? False

James Tam

Know Your Terminology/Watch Operators

- Assignment: =
  - Puts the result of the RHS expression into the memory location on the LHS
  - Examples:

```
Num = 12
PI = 3.14
Area = length * width
```

- Equality: ==
  - Asks a question with a Boolean result
  - Examples:

```
print(1 == 2)
if(age >=18)
```

James Tam

Don't Mix Up Operators

- Asks a question (answer = True so output is 1)

```
if(100 == 100):
    print(1)
else:
    print(2)
```

- Performs assignment
  - Inappropriately attempts an assignment when a Boolean result is expected.

```
if(100 = 100):
    print(1)
else:
    print(2)
```

  - Python's interpretation:
    - Using wrong operator when Boolean result expected.
    - The result of the assignment will not evaluate to a Boolean value.

James Tam

Inappropriate Expressions

- Valid expression but why would you write code this in a real life example?

```
print(1==2)  #Output is false
```

- The resulting expression does not evaluate to something that can be passed as an argument to the function.

```
print(1=2)
```

James Tam

A "Heads Up": Notation

- The approach that was taught at this department for specifying the base use a subscript.
- Examples:
  - Binary to decimal: 000100_2 to _____$_{10}$
  - Hexadecimal to octal: $A1B_{16}$ to _____$_2$
  - Sometimes Hexadecimal includes an embedded '0x'
    - E.g. the above number $A1B_{16}$ is written as 0xA1B

James Tam

WHILE: Write A FOR-Loop Equivalent

```
i = 1
while(i<=3):
    print(i,end="-")
    i = i + 1
print()
```

Sequence 1,2,3

James Tam

WHILE: Write A FOR-Loop Equivalent

```
#While
i = 1
while(i<=3):
    print(i,end="-")
    i = i + 1
print()

#For
for i in range(1,4,1):
    print(i,end="-")
print()
```

Sequence 1,2,3

James Tam

FOR: Write A WHILE-Loop Equivalent

```
for i in range(1,4,1):
    print(i,end="-")
print()
```

Sequence 1,2,3

James Tam

FOR: Write A WHILE-Loop Equivalent

```
#For
for i in range(1,4,1):
    print(i,end="-")
print()

#While
i = 1
while(i<4):
    print(i,end="-")
    i = i + 1
print()
```

Sequence 1,2,3

- **Approach from left there < to continue loop**
- **if counting up (add) then <**

-3 -2 -1 0 1 2 3 4

- **Why "less than"?**
- Sequence is: 1,2,3
- Counting up (approaching from left on the number line for addition).
  - Therefore as long as number in sequence is less than than the end point of 4 continue the sequence.

James Tam

FOR: Write A WHILE-Loop Equivalent

```
for i in range(-4,0,-1):
    print(i,end="-")
print()
```

Loop never runs

James Tam

FOR: Write A WHILE-Loop Equivalent

```
#For
for i in range(-4,0,-1):
    print(i,end="-")
print()

#While
i = -4
while(i > 0):
    print(i,end="-")
    i = i - 1
print()
```

Loop never runs

Loop runs when loop control is greater than stop boundary

Stop boundary

Minus 1 approach from right: subtraction

-3 -2 -1 0 1 2 3

- Why "greater than"?
- Counting down (approaching from right because it's subtraction) e.g. --1,-2-3
  - Therefore as long as number in sequence is greater than the end point continue the sequence.

James Tam

FOR: Write A WHILE-Loop Equivalent

```
for i in range(8,5,1):
    print(i,end="-")
    i = i - 1
print()
```

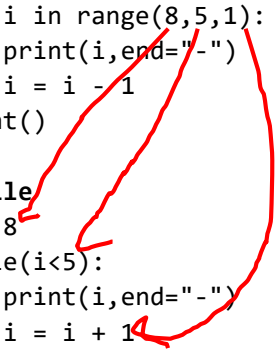
Loop never runs

James Tam

FOR: Write A WHILE-Loop Equivalent

```
#For
for i in range(8,5,1):
    print(i,end="-")
    i = i - 1
print()

#While
i = 8
while(i<5):
    print(i,end="-")
    i = i + 1
print()
```



Loop never runs

-3 -2 -1 0 1 2 3

- Why "less than"?
- Counting up (approaching from left because it's addition) e.g. 1,2,3
  - Therefore as long as number in sequence is less than the end point continue the sequence.

James Tam