*Python Midterm Review – Part 1* Python 3.4.1

1. What function do you use to read a string?
   1. input("Enter a string")
   2. **eval (input ("Enter a string"))**
   3. enter ("Enter a string")
   4. eval (enter("Enter a string"))
2. What is the result of print (1 + 3 \* 2)?
   1. "1 + 3 \* 2"
   2. **7**
   3. 8
   4. "1 + 6"
3. \_\_\_\_\_\_\_ is the code in natural English like language mixed with some program code.
   1. Python program
   2. A Python statement
   3. **Pseudocode**
   4. A flowchart diagram
4. If you enter 1 2 3 in one line, when you run this program, what will happen?

print ("Enter three numbers: ")

number1 = int (input ())

number2 = int (input ())

number3 = int (input ())

# Compute value

value = int((number1 + number2 + number3) / 4)

# Display result

print (value)

1. The program runs correctly and displays 1.5
2. The program runs correctly and displays 1
3. The program runs correctly and displays 1.0
4. The program runs correctly and displays 6/4
5. **The program will have a runtime error on the input**
6. If you enter 1 2 5 one at a time, when you run this program, what will happen?

print ("Enter three numbers: ")

number1 = int (input ())

number2 = int (input ())

number3 = int (input ())

# Compute average

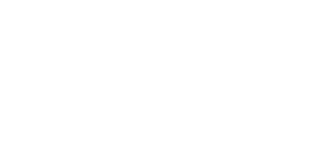
average = int ((number1 + number2 + number3) / 3)

# Display result

print (average)

* 1. The program runs correctly and displays 2.6
  2. **The program runs correctly and displays 2**
  3. The program runs correctly and displays 2.0
  4. The program runs correctly and displays 6/4
  5. The program will have a runtime error on the input

1. You use this symbol to show a decision instruction:



* 1. true
  2. **false**

1. An identifier cannot be a keyword?
   1. **true**
   2. false
2. An identifier can contain digits, but cannot start with a digit?
   1. **true**
   2. false
3. Which of the following is a valid identifier?
   1. $343
   2. **mile**
   3. 9X
   4. 8+9
   5. max-radius
4. Which of the following is a valid identifier?
   1. import
   2. **mile1**
   3. 1MILE
   4. (red)
   5. "red"
5. What will be displayed by the following code?

x = 7 + 3 \* 6 / 2 – 1

print (x)

1. 29
2. **15**
3. 15.0
4. 60
5. What will be displayed by the following code?

x = 1

x = 2 \* x + 1

print (x)

* 1. 0
  2. 1
  3. 2
  4. **3**
  5. 4

1. What will be displayed by the following code?

x = 1

x = x + 2.5

print (x)

* 1. 1
  2. 2
  3. 3
  4. **3.5**
  5. The statements are illegal

1. What is the result of int (45 / 4)?
   1. 10
   2. **11**
   3. 11.25
   4. 12
2. In the expression 45 / 4, the values on the left and right of the / are called \_\_\_\_.
   1. operators
   2. operands
   3. parameters
   4. **arguments**
3. Which of the following expressions will yield 0?
   1. 1 / 2
   2. 1.0 / 2
   3. **1 // 2**
   4. **1.0 // 2**
   5. 1 / 2.0
4. Which of the following expression results in a value 1?
   1. 2 % 1
   2. 15 % 4
   3. 25 % 5
   4. **37 % 6**
5. 25 % 1 is \_\_\_\_\_
   1. 1
   2. 2
   3. 3
   4. 4
   5. **0**
6. What is y displayed in the following code?

x = 1

y = x = x + 1

print ("y is", y)

* 1. y is 0
  2. y is 1 because x is assigned to y first
  3. **y is 2 because x + 1 is assigned to x and then x is assigned to y**
  4. It has a compile error since x is redeclared in the statement int y = x = x + 1

1. Which of the following is equivalent x = 2 % 2 + 2 \* 2 – 2 / 2?
   1. 0.0
   2. 1.0
   3. **3.0**
   4. **3**
   5. 5.0
2. What is the output of this code snippet?

x = 0

while x < 1:

print (x)

x += 1

print (x)

* 1. 0 and 0
  2. 1 and 1
  3. 1 and 0
  4. **0 and 1**

1. What is the result of evaluating 2 + 2 \*\* 3 / 2?
   1. 4
   2. **6**
   3. 4.0
   4. 6.0
2. What is the value of i printed?

j = i = 1

i += j + j \* 5

print ("What is i?", i)

* 1. 0
  2. 1
  3. 5
  4. 6
  5. **7**

1. What is x after the following statements?

x = 1

x \*= x + 1

* 1. x is 1
  2. **x is 2**
  3. x is 3
  4. x is 4

1. What is x after the following statements?

x = 2

y = 1

x \*= y + 1

* 1. x is 1
  2. x is 2
  3. x is 3
  4. **x is 4**

1. To add a value 1 to variable x, you write
   1. 1 + x = x
   2. **x += 1**
   3. x := 1
   4. **x = x + 1**
   5. **x = 1 + x**
2. Which of the following statements are the same?
3. x -= x + 4
4. x = x + 4 – x
5. x = x - (x + 4)
   1. (A) and (B) are the same
   2. **(A) and (C) are the same**
   3. (B) and (C) are the same
   4. (A), (B), and (C) are the same
6. To add number to sum, you write (Note: Python is case-sensitive)
   1. number += sum
   2. number = sum + number
   3. sum = Number + sum
   4. **sum += numbersum = sum + number**
7. Suppose x is 7 and y is 8, what is the output of this code?

if x > y:

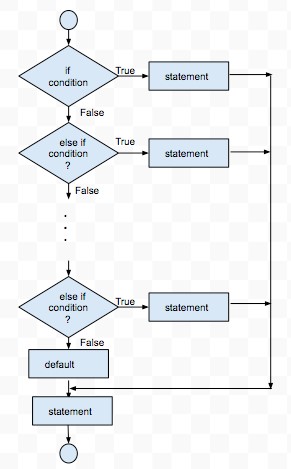
print ("x is ...")

else:

print ("x is ...")

* 1. x is equal to y
  2. **x is smaller than y**
  3. x is bigger than y
  4. 7 is smaller than 8
  5. nothing will be displayed

1. Suppose x is 1. What is x after x -= 1?
   1. **0**
   2. 1
   3. 2
   4. -1
   5. -2
2. When would you use a flowchart like the one below?



* 1. To calculate compound interest
  2. To convert temperatures from Fahrenheit to Celsius
  3. To convert letter grade to string grade
  4. **To convert numeric grade to letter grade**

1. What does this code snippet do?

import random

yesNo = input ("Would you like a random number? y/n ")

while yesNo == "y":

print (random.randrange (1,7))

yesNo = input ("would you like a random number? y/n ")

print ("goodbye")

* 1. **Repeatedly, it prompts the user if she/he wants a random number between 1 and 6 inclusive as long as the user types “y”**
  2. Repeatedly, it prompts the user if she/he wants a random number between 0 and 6 inclusive as long as the user types “n”
  3. Repeatedly, it prompts the user if she/he wants a random number between 1 and 7 inclusive as long as the user types “y”
  4. Repeatedly, it prompts the user if she/he wants a random number between 1 and 7 inclusive as long as the user types “n”

1. What does this instruction print? print (range (0,10))
   1. **[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]**
   2. [1, 2, 3, 4, 5, 6, 7, 8, 9]
   3. range (0, 10)
   4. range (0, 9)
2. What is the output from this code snippet?

for counter in range (100,1, -1):

print (counter)

* 1. It prints integers from 100 to 0
  2. It prints integers from 100 to 1
  3. It prints integers from 99 to 0
  4. It prints integers from 99 to 1
  5. **It prints integers from 100 to 2**

1. What is the output from this code snippet?

sum = 0

for number in range (2,10,2):

sum += number

print ("The sum is ", sum)

* 1. **20**
  2. 30
  3. 28
  4. 44
  5. 54

1. The key word True is \_\_\_\_\_\_\_\_.
   1. A Python Keyword
   2. **A Boolean Identifier**
   3. Same as value 1
   4. Same as value 0
2. Suppose x = 1, y = -1, and z = 1. What will be displayed by the following statement?

if x > 0:

if y > 0:

print("x > 0 and y > 0")

elif z > 0:

print("x < 0 and z > 0")

* 1. x > 0 and y > 0
  2. **x < 0 and z > 0**
  3. x < 0 and z < 0
  4. nothing displayed

1. Analyze the following code:

even = False

if even = True:

print("It is even!")

* 1. The program has a syntax error in line 1 (even = False)
  2. **The program has a syntax error in line 2 if even = True is not a correct condition. It should be replaced by if even == True: or if even:**
  3. The program runs, but displays nothing
  4. The program runs and displays It is even!

1. The following code displays \_\_\_\_\_\_\_\_\_\_\_

temperature = 50

if temperature >= 100:

print("too hot")

elif temperature <= 40:

print("too cold")

else:

print("just right")

* 1. too hot
  2. too cold
  3. **just right**
  4. too hot too cold just right

1. Analyze the following code:

**Code 1:**

if number % 2 == 0:

even = True

else:

even = False

**Code 2:**

even = number % 2 == 0

* 1. Code 1 has compile errors
  2. Code 2 has compile errors
  3. Both Code 1 and Code 2 have compile errors
  4. **Both Code 1 and Code 2 are correct, but Code 2 is better**

1. Which of the Boolean expressions below is incorrect?
   1. True and 3 >= 4
   2. not(x > 0) and (x > 0)
   3. (x > 0) or (x < 0)
   4. **(x not= 0) or (x = 0)**
   5. **(-10 < x < 0)**
2. The order of the precedence (from high to low) of the operators +, \*, /, % is:
   1. **/, %, \*, +**
   2. \*, +, /, %
   3. \*, +, %, /
   4. \*, %, +, /
   5. %, /, +, \*
3. Which asterisk diagram this code displays?

for counter in range(1,12):

for x in range(11,counter,-1):

print ("\*",end='')

print ()

1. A
2. **B**
3. C
4. D

