**Level 1: Basic Math and Strings**

1 b) Expression: 4+7-2+10-6+2; Result: 15

2 b) Expression: 1\*4/1\*5\*3/9; Result: 6.666666666666667

3 a) Expression: 9/3; Result: 3.0

b) Expression: 56/5; Result: 11.2

4) Expression: 9/3; Result: 3

Expression: 56/5; Result: 11

5) a) “==” is not the same as “=” because “=” is used to define or assign the value of a word or string. The double equal sign questions the computer: “is that equal to this thing?”

b) “=” means giving a value to the computer or telling it “this equals that”.

6) a) 5\*5-4+2==100/4-2

True

b) 64\*48+12<45/12-95

False

7) When we type apple in quotes the computer makes it a string. Without the quotes it won’t understand the word apple until it’s been assigned value.

b) Because when we type it in quotations, it is then a string and the program thinks of it as a word or a phrase and not a number.

8) a) “appl” + “e” works because two strings can be added. “apple” - “e” does not work because two strings cannot be subtracted.

b) When we put it in multiplication, Python writes down the number that many times. If we try to divide however, Python does not understand it because a string cannot be divided.

9)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NAME | R | A | G | H | A | V |
| LETTER INDEX | 0 | 1 | 2 | 3 | 4 | 5 |

10) a) It does not print “l” because the computer started counting at 0. So, index 4 in the word “Hello” will be “o”.

b) It prints a space “ “. This is because Python also counts spaces as the index.

11) It says string index is out of range. This means that the word or expression typed does not have that index or it’s too high.

**Level 2: Booleans & Variable**

1) a) Result= 12.0

b) This is because the word “kittens” is not defined and we didn’t give a value to kittens.

2) First, we assigned a new value to puppies i.e. 36. Then we updated the value of “puppies” by taking the value from above and dividing by 3. Finally, the word “puppies” has new value which is 12.

4) First, we assigned a string value to a variable. Now thing value is in the memory of python. Next, we also assigned an integer value to puppies which is 36. At the end we tried adding it but it didn’t work because python is not able to add numbers (integers) with words (strings) so it gave us an error telling us that it is not possible to add an integer to a string.

5) They gave different results because of the order of operations. It calculates the brackets first and then multiplies it by three. Without the brackets it calculated ‘Monday’ times three prior to adding ‘color’.

6) a) Index of ‘r’ in “watermelon” is 4.

b) mynumber = 6; fruit[mynumber-2]

7) “==” is not the same as “=” because “=” is used to define or assign the value of a word or string. The double equal sign questions the computer: “is that equal to this thing?”. “=” means giving a value to the computer or telling it “this equals that”.

8) a) Because Python cannot concatenate objects of different types. “friend” is a string and 5 is an integer.

b) int means integer. It is a whole number.

Str means string. A word or an expression.

9) It is an example of runtime error because Python knows what we are saying but cannot follow the instructions.

10) print (“Raghav” + “ ” + “Bhagat”)

11) a) The value of type(“True”) is string.

b) the value of type(True) is Boolean.

c) The result is different because without the quotation marks True is a Boolean with the quotation marks python thinks of this as a string.

12) A boolean is important because it helps us make decisions about what to do in our code. It can also be used as a tool to determine weather an equation is true or false.

13) This is because the computer wouldn’t be able to understand a ‘Maybe’ data value because the answer can be both true or false. The computer wants to give us the most accurate answer i.e. either True or False.

**Level 3: Lists & Logic**

1) a) i. True ii. False iii. False iv. False

b) Yes, there are more combinations of True/False. We can write True/False as many times but just add the operator “and” each time.

c) It is similar to a math operator because it can also be used like the math operators. Also, that Python gives output after we type “and” operator. It is different because the result for this is either True or False.

2) a) i. True ii. True iii. True iv. False

b) It is similar because it takes input and gives output instantly just like the AND operator. It is different because for this operator only one part has to be true whereas for AND operator both parts have to be True.

3) a) i. False ii. False iii. False iv. True

b) The NOT OR operator gives the opposite output of what OR operator does. It is similar to the AND operator because it can have an infinite number of True or False in them.

4) a) This is because of order of operations. First python does the brackets and then puts ‘not’ in the equation.

b) This is because of order of operations. First python does the brackets and then puts ‘not’ in the equation.

5) a) “Raghav”== “Bhagat” : False

False and 3!=2 : False

True or 5+5

6) myteam = [“Raptors”, “Blue Jays”, “Maple Leafs”]

7) a) The index of the last team is 2.

b) It is an example of runtime error because Python knows what we are saying but cannot follow the instructions.

9) mynumber = 24

if mynumber == 24:

print ("Hi Alfred")

10) myname="Raghav"

if myname == "Ginger":

print("Hi Ginger!")

else:

print("Raghav")

11) myname=input()

if myname == "Dylan":

print("Hi Dylan!")

elif myname == "Raj":

print("Hi Raj!")

elif myname == "Moulik":

print("Hi Moulik")

elif myname == "Gurjap":

print("Hi Gurjap!")

else:

print("Who are you?!?")