ASSIGNMENT - 1

Q1. Which of the following are operators, and which are values?

a) *

b)'hello'

c)-88.8

d)-

e)/

f) +

g)5

- ✓ Ans:

Operators = [a,d,e,f] -> Arithmetic

Values = [b,c,g]

Q2.Which of the following is a variable, and which is a string?

a)spam

b)'spam'

- ✓ Ans :

Variable = (a) spam

'spam' = (b) string, bcz of ''

Q3.Name three data types

- ✓ Ans :

Text Type : str

Numeric Types : int, float, complex

Sequence Types : list, tuple, range

Mapping Type : dict

Set Types : set, frozenset

Boolean Type : bool

Binary Types : bytes, bytearray, memoryview

None Type : NoneType

Q4.. What is an expression made up of? What do all expressions do?

- ✓ Ans :

An expression made up of values,variable and operators

The all expression getting the values from variable and perfome the operation with operator in single value

like :

a = 10

b =20

result = a+b

print(result) #output = 30

Q5. This chapter introduced assignment statements, like spam = 10. What is the difference between an expression and a statement?

- ✓ Ans:

An expression evaluates to a single value. A statement does not.

Q6.. What does the variable bacon contain after the following code runs?

a) bacon = 20

b) bacon + 1

- ✓ Ans:

bacon contain 20 after code run

because the value increment by 1 but does not store in bacon so it's remained same

---

Q7. What should the following two expressions evaluate to?

a)'spam' + 'spamspam'

b)'spam' * 3

- ✓ Ans:

a = 'spamspamspam' #here the two string combines in single string

b = 'spamspamspam' #string multiplies with (*)expression

Q8.Why is eggs a valid variable name while 100 is invalid?

Ans :

Bcz the egg starting with character and does not contain any space [btwn character] and special character and number, where 100 is number so as variable it is invalid

---

Q9.What three functions can be used to get the integer, floating-point number, or string version of a value?

- ✓ Ans:

For integer : int()

float : float()

string : str()

Q10.Why does this expression cause an error? How can you fix it?

'I have eaten ' + 99 + ' burritos.'

- ✓ Ans:

back of the string concinnated other strings with + [operator] but here 99 is number so its cause error

we can fix it with,

'I have eaten ' + '99' + ' burritos.'

ASSIGNMENT - 2

Q1. What is []?

- ✓ Ans:

This is empty list which indicates with [] bracket

Q2. How would you assign the value 'hello' as the third value in a list stored in a variable named spam? (Assume spam contains [2, 4, 6, 8, 10].)

- ✓ Ans:

```
spam = [2,4,6,8,10]

spam.insert(2,"hello")

# output [2, 4, 'hello', 6, 8, 10]
```

Q3. For the following three questions, let’s say spam contains the list ['a', 'b', 'c', 'd'].

3. What does spam[int(int('3' * 2) // 11)] evaluate to?

- ✓ Ans:

```
spam = ['a','b','c','d']
spam[(int('3'*2)//11)]
# output : ‘d’
```

[Explanation] : int('3'*2) so output like 33, 33 divide by 11 output is 3, spam[3] = d, Result = 'd'

Q4. What does spam[-1] evaluate to?

- ✓ Ans:
- ✓ spam = ['a','b','c','d']
  spam[-1]
  #output = 'd'

Q5. What does spam[:2] evaluate to?

- ✓ Ans:

  spam = ['a','b','c','d']
  spam[:2]
  #output = ['a','b']

Q6. For the following three questions, let's say bacon contains the list [3.14, 'cat', 11, 'cat', True].

6. What does bacon.index('cat') evaluate to?

- ✓ Ans:

  balcon = [3.14,'cat',11,'cat',True]
  balcon.index('cat')
  #output : 1 <= return the index value of cat

Q7. What does bacon.append(99) make the list value in bacon look like?

- ✓ Ans:

  balcon = [3.14,'cat',11,'cat',True]
  bacon.append(99)
  # output : [3.14,'cat',11,'cat',True,99]

Q8. What does bacon.remove('cat') make the list value in bacon look like?

- ✓ Ans:

  balcon = [3.14,'cat',11,'cat',True]
  balcon.remove('cat')
  #output : [3.14, 11, True, 99]

Q9. What are the operators for list concatenation and list replication?

- ✓ Ans:

  For list concatenation operator is: +
  For list replication operator is: *

Q10. What is the difference between the append() and insert() list methods?

- ✓ Ans:

  append() will add value end of the list while insert add value anywhere in list

Q11. What are two ways to remove values from a list?

- ✓ Ans:

  pop(), del(delete) and remove()

Q12. Name a few ways that list values are similar to string values.

- ✓ Ans:
  - Both can be passed in len() function
  - Have index and slices
  - Used in for loop
  - Concatenated and replicated both have operation
  - Both use in and not in operators

Q13. What is the difference between lists and tuples?

- ✓ Ans :

  List are mutable, you can add values remove it or changed it, indicates with [] symbol
  Tuple are immutable, you can't change it, it indicates with () symbol

Q14. How do you type the tuple value that has just the integer value 42 in it?

- ✓ Ans:

  (42,)

Q15. How can you get the tuple form of a list value? How can you get the list form of a tuple value?

- ✓ Ans:

  To get tuple use tuple() function and to get list use list() function

Q16. Variables that "contain" list values don't actually contain lists directly. What do they contain instead?

- ✓ Ans:

  They contain the references to list value.
  [Explanation]
  When you assign a list to a variable, you are actually assigning a list *reference* to the variable. A reference is a value that points to some bit of data, and a list reference is a value that points to a list.

```
spam = [0, 1, 2, 3, 4, 5]

>>> cheese = spam
>>> cheese[1] = 'Hello!'
>>> spam
      [0, 'Hello!', 2, 3, 4, 5]
>>> cheese
      [0, 'Hello!', 2, 3, 4, 5]
>>>id(spam) , id(cheese)
   (2231400524608, 2231400524608)
```

This might look odd to you. The code changed only the cheese list, but it seems that both the cheese and spam lists have changed.
When you create the list spam, you assign a reference to it in the spam variable. But the next line cheese copies only the list reference in spam to cheese, not the list value itself. This means the values stored in spam and cheese now both refer to the same list. There is only one underlying list because the list itself was never actually copied. So, when you modify the first element of cheese, you are modifying the same list that spam refers to.
As we can see the both lists have same id value.

Q17. What is the difference between copy.copy() and copy.deepcopy() ?

- ✓ Ans:

copy.copy() will shallow copy the list while copy.deepcopy() duplicate any list inside the list(nested list).
[Explanation]

```
x = [["hi"],11,12,13]
y = copy.copy(x)
print("before changing value")
print(x)
print(y)
x[0][0] = "hello"
print("after changing value")
print(x)
print(y)
```

#output

```
before changing value
[['hi'], 11, 12, 13]
[['hi'], 11, 12, 13]
after changing value
[['hello'], 11, 12, 13]
[['hello'], 11, 12, 13]
```

- In copy, the both list reference are same so if you change original it will also impact on copy list. \

```
x = [["hi"],11,12,13]
y = copy.deepcopy(x)
print("before changing value")
print(x)
print(y)
x[0][0] = "hello"
print("after changing value")
print(x)
print(y)
```

#output

```
before changing value
[['hi'], 11, 12, 13]
[['hi'], 11, 12, 13]
after changing value
[['hello'], 11, 12, 13]
[['hi'], 11, 12, 13]
```

- In deepcopy, the new object has make new copy of list when you change the nested list value of original list it will not impact on deepcopy object

---

ASSIGNMENT - 3

Q.1 What does the code for an empty dictionary look like?

- ✓ Ans:

  {}

Q2. What does a dictionary value with a key 'foo' and a value 42 look like?

- ✓ Ans:

  d = {‘foo’ : 42}

Q3. What is the main difference between a dictionary and a list?

- ✓ Ans:
  - Value stores in dictionary are unorder while in list its order.
  - For dictionary you have to define value with key, while in list simply you can define value.

Q4. What happens if you try to access spam['foo'] if spam is {'bar': 100}?

- ✓ Ans:

  KeyError: ‘foo’

Q5. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys()?

- ✓ Ans:

  No difference, both return True value

  >>>spam = {'cat' : 1}

  >>>'cat' in spam

  True

  >>>'cat' in spam.keys()

  True

Q6. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()?

- ✓ Ans:

  Well here spam.values check whether particular value exists in dictionary or not.

  >>>spam = {'cat' : 1}

  >>>'cat' in spam

  True

  >>>'cat' in spam.values()

  False

Q7. What is a shortcut for the following code?

```
if 'color' not in spam:
    spam['color'] = 'black'
```

- ✓ Ans:

  spam.setdefault(‘color’,’black’)

Q8. What module and function can be used to “pretty print” dictionary values?

- ✓ Ans:

  import pprint

  pprint.pprint()

ASSIGNMENT - 4

Q1. What are escape characters?

- ✓ Ans:

  Escape characters use for string to overcome the error, insert characters that illegal in a string you can use escape characters for legal them and use in code.

  Like: \, \\, \n, \t, \r

Q2. What do the \n and \t escape characters represent?

- ✓ Ans:

  \n for new line and \t for tab the characters or sentence in string.

Q3. How can you put a \ backslash character in a string?

- ✓ Ans:

  With use ➔ \\

Q4. The string value "Howl's Moving Castle" is a valid string. Why isn’t it a problem that the single quote character in the word Howl's isn’t escaped?

- ✓ Ans:

  Because of doble quotes (“”) beginning in the string so single quote will be fine inside it.

Q5. If you don’t want to put \n in your string, how can you write a string with newlines in it?

- ✓ Ans:

  Multiline string ➔ """ """ or ''' '''

Q6. What do the following expressions evaluate to?

a) 'Hello world!'[1]
b) 'Hello world!'[0:5]
c) 'Hello world!'[:5]

d) 'Hello world!'[3:]

✓ Ans:

a) ‘e’
b) ‘Hello’
c) ‘Hello’
d) ‘lo world!‘

Q7. What do the following expressions evaluate to?

a) 'Hello'.upper()
b) 'Hello'.upper().isupper()
c) 'Hello'.upper().lower()

✓ Ans:

a) ‘HELLO’
b) True
c) hello

Q8. What do the following expressions evaluate to?

a) 'Remember, remember, the fifth of November.'.split()
b) '-'.join('There can be only one.'.split())

✓ Ans:

a) ['Remember,', 'remember,', 'the', 'fifth', 'of', 'November.']
b) 'There-can-be-only-one.'

Q9. What string methods can you use to right-justify, left-justify, and center a string?

✓ Ans:

rjust(), ljust(), center()

Q10. How can you trim whitespace characters from the beginning or end of a string?

✓ Ans:

For left of string use: lstrip()
For right of string use: rstrip()

ASSIGNMENT – 5

Q1. What are the two values of the Boolean data type? How do you write them?

✓ Ans:

True and False ← write like them

Q2. What are the three Boolean operators?

✓ Ans:

AND, OR, NOT also AND NOT

Q3. Write out the truth tables of each Boolean operator (that is, every possible combination of Boolean values for the operator and what they evaluate to).

✓ Ans:

True and True ➔ True
True and False ➔ False
False and True ➔ False
False and False ➔ False
True or False ➔ True
True or True ➔ True
False or True ➔ True
False or False ➔False
Not True ➔ False
Not False ➔ True

Q4. What do the following expressions evaluate to?

1)(5 > 4) and (3 == 5)

2) not (5 > 4)

3) (5 > 4) or (3 == 5)

4) not ((5 > 4) or (3 == 5))

5) (True and True) and (True == False)

6) (not False) or (not True)

✓ Ans:

1) False
2) False
3) True
4) False
5) False
6) True

Q.5 What are the six comparison operators?

- ✓ Ans:
  ==, !=, <, >, <= and >=

Q6. What is the difference between the equal to operator and the assignment operator?

- ✓ Ans:
  Equal (==) operator evaluates two values and return Boolean value True or False
  While assignment operator (=) stores the value in variable

Q7. Explain what a condition is and where you would use one.

- ✓ Ans:
  I. Conditional statements, expressions, or simply conditionals are features of programming languages that tell the computer to execute certain actions, provided certain conditions are met.
  II. These decisions are made if and only if the pre-stated conditions are either **true** or **false**, depending on the functions the programmer has in mind.

ASSIGNMENT – 7

Q1. Write a Python program to find maximum between two numbers.

- ✓ Program:

```
num1 = int(input("Enter the 1st number:"))
num2 = int(input("Enter the 2nd number:"))
print(f"{num1} is maximum") if num1>num2 else print(f"{num2} is maximum")
l = [num1 if num1>num2 else num2]
print("maximum is:",l)
```

Q2. Write a Python program to find maximum between three numbers.

- ✓ Program:

```
num1 = int(input("Enter the 1st number:"))
num2 = int(input("Enter the 2nd number:"))
num3 = int(input("Enter the 3rd number:"))
if num1>num2 and num1>num3:
    print("Maximum number is",num1)
elif num2>num1 and num2>num3:
    print("Maximum number is",num2)
else:
    print("Maximum number is:",num3)

#SECOUND WAY WITH sort function
l = [num1,num2,num3]
l.sort()
print("Maximum number is:",l[-1])
```

Q3. Write a Python program to check whether a number is negative, positive or zero.

✓ Program:

```
num = int(input("Enter the number:"))
if num>0:
    print(f"{num} is positive")
elif num==0:
    print(f"{num} is Zero")
else:
    print(f"{num} is negative")
```

Q4. Write a Python program to check whether a number is divisible by 5 and 11 or not.

✓ Program:

```
num = int(input("Enter the number:"))
if (num%5==0) and (num%11==0):
    print(f"{num} is divide by 5 and 11")
else:
    print("Number not divided")
```

Q5. Write a Python program to check whether a number is even or odd.

✓ Program:

```
num = int(input("Enter the number:"))
if (num%2==0):
    print(f"{num} is even")
else:
    print("Number is odd")
```

Q6. Write a Python program to check whether a year is leap year or not.

✓ Program:

```
year = int(input("Enter the number:"))
if (year%4==0 and year%100!=0) or (year%400==0 and year%100==0):
    print(f"{year} is leap year")
else:
    print(f"{year} not leap year")
```

Q7. Write a Python program to check whether a character is alphabet or not.

✓ Program:

```
ch = input("Enter the string:")
if ch.isalpha():
    print("Alphabet")
else:
    print("No alphabet")
```

Q8. Write a Python program to input any alphabet and check whether it is vowel or consonant.

✓ Program:

```
ch = input("Enter any alphabet:")
vowel = ['a','e','i','o','u','A','E','I','O','U']
if ch in vowel:
    print("{} is vowel".format(ch))
else:
    print("{} is consonant".format(ch))
```

Q9. Write a Python program to input any character and check whether it is alphabet, digit or special character.

✓ Program:

```
st = input("Enter the input:")
if st.isalpha():
    print("Alphabet")
elif st.isdigit():
    print("Digit")
else:
    print("special character")
```

Q10. Write a Python program to check whether a character is uppercase or lowercase alphabet.

✓ Program:

```
alp = input("Enter the alphabet")
if alp.isupper():
    print("{} uppercase".format(alp))
elif alp.islower():
    print("{} lowecase".format(alp))
```

Q11. Write a Python program to input week number and print week day.

✓ Program:

```
weekdays = { 1:'sunday',
             2:'monday',
             3: 'Tuesday',
             4: 'wednesday',
             5: 'Thursday',
             6: 'Friday',
             7: 'Saturday'}
print("*"*10+"Week starts from Sunday"+"*"*10)
week_number = int(input("Enter the week number:"))
print("Week day is:",weekdays[week_number])
```

Q12. Write a Python program to input month number and print number of days in that month.

✓ Program:

```
month = {1:31,2:28,3:31,4:30,5:31,6:30,7:31,8:31,9:30,10:31,11:30,12:31}
num = int(input("Enter the month number:"))
if num == 2:
    print(f"Days in month {month[num]} or 29 if year is leap year")
else:
    print(f"Days in month {num} is:{month[num]}")
```

Q13. Write a Python program to count total number of notes in given amount.

✓ Program:

```
amount = int(input("Enter the amount:"))
notes = (2000,500,200,100,50,20,10,5,2,1)
for i in notes:
    count = amount//i
    print(f"Notes of {i} \t count = {count}")
    amount = amount%i
```

Q14. Write a Python program to input angles of a triangle and check whether triangle is valid or not.

✓ Program:

```
angle_1 = int(input("Enter the angle 1:"))
angle_2 = int(input("Enter the angle 2:"))
angle_3 = int(input("Enter the angle 3:"))
su = angle_1 + angle_2 +angle_3
if su == 180:
    print("Triangle is Valid")
else:
    print("Traingle is not Valid")
```

Q15. Write a Python program to input all sides of a triangle and check whether triangle is valid or not.

✓ Program:

```
side_1 = int(input("Enter the side 1:"))
side_2 = int(input("Enter the side 2:"))
side_3 = int(input("Enter the side 3:"))
if side_1+side_2>side_3 and side_1+side_3>side_2 and side_2+side_3>side_1:
    print("Triangle is Valid")
else:
    print("Triangle is not valid")
```

Q16. Write a Python program to check whether the triangle is equilateral, isosceles or scalene triangle.

✓ Program:

```
side_1 = int(input("Enter the side 1:"))
side_2 = int(input("Enter the side 2:"))
side_3 = int(input("Enter the side 3:"))

if side_1==side_2==side_3:
    print("Triangle is equilateral")
elif side_1==side_2 or side_2==side_3 or side_1==side_3:
    print("Triangle is isosceles")
else:
    print("Triangle is scalene")
```

Q17. Write a Python program to calculate profit or loss.

✓ Program:

```
orignal_prize = int(input("enter the base value:"))
sell_value = int(input("Enter the sell value:"))
if sell_value>orignal_prize:
    profit = sell_value-orignal_prize
    print("Profit =",profit)
elif sell_value<orignal_prize:
    loss = orignal_prize-sell_value
    print("Loss =",loss)
else:
    print("no profit no loss")
```

Q18. Write a Python program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:

Percentage >= 90% : Grade A

Percentage >= 80% : Grade B

Percentage >= 70% : Grade C

Percentage >= 60% : Grade D

Percentage >= 40% : Grade E

Percentage < 40% : Grade F

✓ Program:

```
subjects = ['Physics', 'Chemistry', 'Biology', 'Mathematics','Computer']
marks=[]
for i in subjects:
    inp = int(input(f"Enter the {i} marks:\t"))
    marks.append(inp)
per = sum(marks)/(len(subjects)*100)*100
print("Student got: {per:.2f}%".format(per=per))
if per >= 90:
    print("Grade A")
elif per >= 80:
    print("Grade B")
elif per>=70:
    print("Grade C")
elif per>=60:
    print("Grade D")
elif per >= 40:
    print("Grade E")
else:
    print("Grade F")
```

Q19. Write a Python program to input basic salary of an employee and calculate its Gross salary according to following:

Basic Salary <= 10000 : HRA = 20%, DA = 80%

Basic Salary <= 20000 : HRA = 25%, DA = 90%

Basic Salary > 20000 : HRA = 30%, DA = 95%

- ✓ Program:

```
s=str("EMPLOYEE-SALARY")
print(s)
print("="*len(s))
Basic_Salary = int(input("Enter the basic salary : "))
if Basic_Salary <= 10000:
    HRA = 0.20*Basic_Salary
    DA  = 0.80*Basic_Salary
elif Basic_Salary <= 20000:
    HRA = 0.25*Basic_Salary
    DA  = 0.90*Basic_Salary
else:
    HRA = 0.30*Basic_Salary
    DA  = 0.95*Basic_Salary
Gross_Salary = Basic_Salary+HRA+DA
print("Gross Salary is:",Gross_Salary)
```

Q20. Write a Python Program to input electricity unit charges and calculate total electricity bill according to the given condition:

For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit

For next 100 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill

- ✓ Program:

```
print("Welcome to Electricity Board")
unit = float(input("Enter the unit value:"))
if unit <=50:
    bill = unit*(0.50)
elif unit>50 and unit<=150:  #for next 100 unit, btwn 50 to 150 unit
    initial_unit = (50*0.50) #as we have to add first 50 unit charge
    bill = initial_unit + ((unit-50)*0.75)
elif unit >150 and unit<=250: #for next 100 unit, btwn 150 to 250 unit
    initial_unit = (50*0.50)  #as we have to add first 50 unit charge
    initial_unit_2=(100*0.75) #add  100 unit charge
    bill = initial_unit + initial_unit_2 +((unit-150)*1.20)
else:                         #For all unit>250
    initial_unit = (50*0.50)  #as we have to add first 50 unit charge
    initial_unit_2=(100*0.75) #add next 100 unit charge
    initial_unit_3 = (100*1.20) #add another next 100 unit charge
    bill = initial_unit + initial_unit_2 +initial_unit_3 +((unit-250)*1.50)
print("Bill without surcharge :" ,bill)
Add = bill*20/100
print("Total Bill in Rs.= ",bill+Add)
#Note : keep in mind that like if our unit is = 200 then first 50 unit charge 0.50/unit, secound 100 charge 0.75/unit will be
#add in your calculation and remaining 50 unit out of 200 will be charge 1.20/unit.
```