

# \* ASSIGNMENT - I \*

1) Develop a python program to perform the bitwise operations on 60 and 13

## ALGORITHM:

Step 1: Define two variables, 'a' and 'b' and assign the values 60 and 13 to them

Step 2: Perform a bitwise AND operation between 'a' & 'b' and store the result in 'c'

Step 3: Perform a bitwise OR operation between 'a' & 'b' and store the result in 'd'

Step 4: Perform a bitwise EXOR operation between 'a' & 'b' and store the result in 'e'

Step 5: Perform a bitwise NOT operation on 'a' & 'b' separately and store the results in 'f' & 'g'

Step 6: Print the results of all the bitwise operations.

Step 7: End.

2) Perform the following operations on the given list.

A = ['abcd', 456, 2.23, 'john', 70.2]

B = [123, 'john']

- Print list A & B
- Print the first element of list A
- Print the elements starting from 1-h of list A
- Print the list B for two times
- Join the list A & B
- Replace all the elements of list B
- Delete the second element from list A
- Insert the new element in the list B

#### ALGORITHM:

Step 1: Define two lists, 'A' & 'B' with the given elements.

Step 2: Print the contents of both lists, 'A' & 'B'.

Step 3: Print the first element of list 'A'

Step 4: Print the elements from index 1 to h (inclusive) of list A

Step 5: Print list B repeated twice

Step 6: Concatenate the two lists and print the outcome

Step 7: Replace any element of list B with new values

Step 8: Delete the second element from list A

Step 9: Insert a new element into list B.

Step 10: Print the updated list A & list B.

Step 11: End

3) Develop a python program that prints the reference of the variable  $a=10$  &  $b=10$  using the identity operator

ALGORITHM:

Step 1: Define variables 'a' & 'b' and assign the value 10 to both of them

Step 2: Use the identity operator 'is' or 'is not' to check if 'a' & 'b' have the same references

Step 3: If they have the same reference , print "TRUE" otherwise , print "FALSE".

Step 4: End

Develop a python program on the nested list.

ALGORITHM:

Step 1: Define a nested list with multiple lists inside

Step 2: To access an element in the nested list, specify the row & column indices.

Step 3: Print the element at a specific row & columns in the nested list.

Step 4: End

5) Illustrate how lists are different from the tuples.

ALGORITHM:

Step 1: Define a Tuple 't'

Step 2: Define a list 'l'

Step 3: Print the list 'l' & Tuple 't'

Step 4: Perform the following operations on the tuple

Step 4.1: Count the elements & save it to variable 'c'

Step 4.2: Append some values to the tuple & save it to tuple 't<sub>2</sub>'

Step 4.3: Print 'c' and 't<sub>2</sub>'

Step 4.4: Print minimum value of Tuple 't'

Step 4.5: Print maximum value of Tuple 't'

Step 4.6: Print length of the Tuple t.

Step 5: End

6) Develop a python program to calculate the gross salary of the employee & provide the bonus of Rs. 5000 to all the employees having their basic salary greater than Rs. 20000

#### ALGORITHM:

Step 1: Define variable 'bs' for basic salary

Step 2: Calculate 'da' & 'hra'

Step 3: check if the basic salary is greater than Rs. 20000

Step 3.1: If the basic salary 'bs' is greater than Rs. 20000  
add a bonus to the gross salary 'gs' of Rs. 5000

Step 3.2: Otherwise, do not add the bonus to the  
gross salary 'gs'

Step 4: Print gross salary 'gs'

Step 5: End

7) Develop a python program that apply the discount of 15% on purchase of items above Rs. 1000

ALGORITHM:

Step 1: Define the variable 'cost'

Step 2: Input the purchase amount from the user

Step 3: Check if the purchase amount is greater than Rs. 1000

Step 3.1: If it is greater than Rs. 1000, calculate a discounted amount by applying a 15% discount

Step 3.2: Otherwise, assign the 'cost' value to 'total'

Step 4: Assign discounted amount, if calculation is done to 'total'

Step 5: Print the value of 'total'.

Step 6: End

8) Develop a python program to calculate the SGPA for the courses of IV semester.

ALGORITHM:

Step 1: Input marks of subjects to variables 'jp', 'ada', 'mc', & 'dmgt'

Step 2: Input credits of the subjects to variables 'a', 'b', 'c' & 'd'.

Step 3: Add the credits of the subjects to variable 'creds'

Step 4: calculate the SGPA of the subject for a 9.0 grade

Step 5: Print the SGPA

Step 6: End

a) Develop a python program to find the largest among three numbers.

ALGORITHM:

Step 1: Input three numbers 'a', 'b' & 'c' from the user

Step 2: Check if 'a' is greater than 'b' and 'c'

Step 2.1: If it is, print 'a'

Step 2.2: Otherwise, go to step 3

Step 3: Check if 'b' is greater than 'a' and 'c'

Step 3.1: If it is, print 'b'

Step 3.2: Otherwise, go to step 4.

Step 4: Print 'c'

Step 5: End

i) 10) Write a python program to calculate the gross salary of 10 employees

ALGORITHM:

Step 1: Define variables for 'basic salary' - 'bs'

Step 2: Use 'for-loop' to calculate the gross salary 'gs'  
for each employee

Step 3: Terminate the loop at 10<sup>th</sup> employee.

Step 4: Print each answer of 'gs' orderly.

Step 5: End

1.1) Write a python program that prints the sum of all even & odd numbers that are between two given numbers.

ALGORITHM:

Step 1: Input the numbers 'a' & 'b' representing the start & end of the numbers.

Step 2: Initialize variables 'total-even' & 'total-odd' to zero

Step 3: Use 'if' statement to check the limit, if it is ascending or descending

Step 3.1: Use 'while' loop to determine the even numbers and add those numbers to 'total-even'

Step 3.2: Terminate the loop after the last number.

Step 3.3: Otherwise, go to step 4

Step 4: If it is descending, then execute the following

Step 4.1: use 'while' loop to determine the even numbers and add those numbers to 'total-even'

Step 4.2: Terminate the loop after the last number & go to step 5

Step 5: Repeat steps '3' & '4' for odd numbers.

Step 6: Print 'total-even' & 'total-odd'

Step 7: End

1. Q) With example program demonstrate the working of Range function.

ALGORITHM:

Step 1: Use the 'range' function to generate a sequence of numbers from 1 to 10 (inclusive)

Step 2: Use a 'for' loop to iterate through the numbers generated by the 'range' function.

Step 3: Inside the loop, print each number.

Step 4: End

- 1) Develop a python program that demonstrate the working of a function with arguments & with return value.

ALGORITHM:

Step 1: Define a function 'add-numbers' that takes two arguments, 'x' & 'y'

Step 2: Inside the function, calculate the sum of 'x' & 'y'

Step 3: Use the 'return' statement to return the result from the function.

Step 4: Call the 'add-numbers' function with two numbers as arguments.

Step 5: Capture the returned value & print it.

Step 6: End

2) With python program illustrate the function calling another function.

ALGORITHM:

Step 1: Define a function 'greet' which takes 'name' as an argument.

Step 2: Define another function 'welcome' that takes another 'name' as an argument.

Step 3: Call the 'greet' function & capture the result with the name

Step 4: Call the 'welcome' function & capture the result with another name

Step 5: Print the final squared - up result.

Step 6: End

3) write a python program that illustrate the working of function without argument & with return value.

ALGORITHM:

- Step 1: Define a function 'get-pi'
- Step 2: return the value '3.14159' to the function
- Step 3: call the function 'get-pi'
- Step 4: print the value of the function.
- Step 5: End

4) Write python program that demonstrates the function with positional arguments.

ALGORITHM:

Step 1: Define a function 'calculate-power' with arguments 'base' & 'exponent'

Step 2: calculate the 'result' of the power of the arguments & return the value.

Step 3: call the function and pass arguments

Step 4: Print the result.

Step 5: End

5) Write python program that demonstrates the function with default argument.

ALGORITHM:

- Step 1: Define a function 'greet' that accepts two arguments, where one of the arguments is declared as "Hello"
- Step 2: return a message in the function ,using the arguments.
- Step 3: Call the function and pass a name in the argument .
- Step 4: Print the message .
- Step 5: End

6) Illustrate the process of creating an object using python program

ALGORITHM:

Step 1: Define a class 'MyClass'

Step 2: Define the attribute 'name' in the class, inside a method '-init-'

Step 3: Create an object from the class 'MyClass' using the constructor.

Step 4: Access the function in the class by the object.

Step 5: End

7) Demonstrate the working of empty class

ALGORITHM:

Step 1: Define an empty class 'EmptyClass' with no methods or attributes.

Step 2: End.

- 9) 8) Define a class student & with help of object define the function  
getdata() & displaydata()
- ALGORITHM:
- Step 1: Define a class 'Student'
- Step 2: Inside the class , define attributes such as 'name', & 'roll'  
& methods such 'getdata()' & 'displaydata()' .
- Step 3: Create an object of the 'Student' class .
- Step 4: Call the ' getdata()' method to input student  
information .
- Step 5: Call the ' displaydata()' method to display the student  
information .
- Step 6 : End .

9) Write a python program to find the area of rectangle.

ALGORITHM:

Step 1: Define a function 'calculate-area' that takes two arguments, 'length' & 'breadth'

Step 2: Inside the function, calculate the area of the rectangle, using the formula 'length \* breadth'

Step 3: Print the area.

Step 4: End

(o) Analyze the importance of 'Self' as a default positional argument in functions.

ALGORITHM:

Step 1: Define a class 'MyClass' with two methods '-init-' & 'increment', with arguments 'value' & 'amount', respectively

Step 2: Create objects for the methods & call the methods by passing arguments in them

Step 3: Print the 'value' & 'amount' from the class

Step 4: End

Q1) Develop a python program that demonstrates the working of '`__init__`' function.

ALGORITHM:

Step 1: Define a class 'MyClass'

Step 2: Inside the class, define the '`__init__`' method that is for the initialization and has a 'value' as an argument.

Step 3: Create an object for the class 'MyClass' & call the function by passing an argument.

Step 4: Print the value passed by the object.

Step 5: End.

12) Develop a python program that assigns the values to the attributes of student class using '`-init-`' function.

ALGORITHM:

Step 1: Define a class 'student'

Step 2: Inside the class , define attributes such as 'name' & 'roll' & an '`-init-`' method to assign values to these attributes during object creation.

Step 3: Create an object of the 'student' class & access the attributes to verify that they are initialized correctly .

Step 4: End .