① What is computer programming? Explain the types of programming languages?

Ans → Computer programming is a step by step process of designing and developing various sets of computer programms to accomplish a specific computing outcome.

- The purpose of a computer programming is to find a sequence of instruction to solve a specific problem on a computer.

- Computer programming is very easy if it is appropriately managed. There are so many programming languages so to finalizing the right programming language is not an easy task.

- There are two types of programming language, which can be categorized into the following ways.

1. Low level language.
2. High level language.

1. low level language.

- This is the most understanable language by the computer to perform its function. This can be further categorized into:

a] Machine learning (ML).

Machine language is consist of strings of binary number. (i.e. 0s and 1s) and it is the only one language the processor directly understands. Machine language has an merit of very fast execution speed.

10/01/2023 19:18

- Merits.

- It is directly understanded by the processor so have faster execution time since the program written in this language, the processor so has faster execution time since the program written in this language is not needed to be translated.

- Demerits.

- It is very difficult to program using ICL since all the instruction are to be represented by 0s and 1s.

- Use of this language makes programming time consuming.

b] Assembly languages:

- Assembly languages is also known as low-level language because to design a programme the programmer requires all the detailed knowledge about the Hardware specification. This language uses mnemonics code (symbolic operation code like 'ADD' for Addition). in place 0s and 1s.

Merits:

- It is makes programming easier than HXL since it uses mnemonic code for programming. Eg: ADD for addition, SUB for substraction DIV for division.

- It makes programming process faster
- Error can be identified much easily than compared to ICL.

Demerits:

- Programm written in this language is not directly understood by the machine or computer so translation should be used.
- It is hardware dependent language so programmers are forced to thinks in terms of computer architecture. Being a machine dependent language, written in this language are very less and or not portable

2] Describe the programming principles:-

1] Keep it simple.

- It means you should be writing scale code as simple as possible. Don't get caught up in trying to be over-clever or showing off with 4 paragraph of advance code. If you can write a script in one line write it in one line.

Eg.
```
function addnumbers (num1, num2).
{
    return num1 + num2
}
```

2] Write Dry Code.

- The Don't Repeat yourself principle means, mainly, not repeating code. It's common coding mistakes when writing code, avoid duplication of data or. If you've ever copied and pasted code within or logic It's not dry code.
- Take a look at this script:

```
{
    number = number + 1;
    number = number + 2;
    number = number + 3;
    number = number + 4;
    number = number + 5;
    return number;
}.
```

3) Open | closed.

- This Principle means you should aim to make your code open to extension but closed to modification. This is an important principle

when releasing a library or framework that others will use.

- For example; suppose you're maintaining a CRUI framework. You could release for coders to directly modify and integer your released code.

4] Single Responsibility;

- The single responsibility principle says that every class or module in a program should only provide one specific functionality. As Robert C. Martin put it, "A class should have only reason to change.

5) Seperation of concerns.

The Seperation of Concerns principle is an abstract version of the single responsibility principle. This idea states that the program should be designed with different containers and these containers should not have access to each other.

A well known example of this is the model view controller design. MVC seperates a program should be desined with the logic, and what the page displays. Variations of MVC are common in todays most popular web framework.

6] You Aren't going to need it (YAGNI).

This principle means you should never code for functionality on the chance that you may used in the future. Don't try and solve 4 problem that Doesn't exist.

7] Document your code

Any Senior developer will stress the importance of documenting your code with proper comment. All languages often then and you should make it a habit to write them. Leave comment to explain objects enhance variable definitions and make function easier to understend.

10/01/2023 19:18

8] Refactor

- It's hard to accept, but your code isn't going to be perfect the first time. Refactorising code means receiving your code and looking for ways to optimize it. Make it more efficient while keeping the result exactly the same

9] Clean code at all costs.

- Don't try to pack a ton of logic into on line. Leave clear instruction in comments and documents. If your code is easy to read it will be easy to maintain.

Q-3] Differentiate compiler and interpreter.

Compiler	Interpreter
A compiler is a programm which converts the entire source code of programming lang. into executable machine code for a CPU.	Interpreter takes the source program and run it line by line, translating each line as it comes to it.
Compiler takes large amount of time to analyze the entire source code but the overall execution time of the programm is comparatively faster	Interpreter takes a less time to analyze the source code but the overall execution time of programm is slower.
Compiler generates the error message only after scanning the whole program so debugging is comparatively hard as the error can be present any where in the programme	debugging is easier as it continuously translates the program until the error is met

10/01/2023 19:18

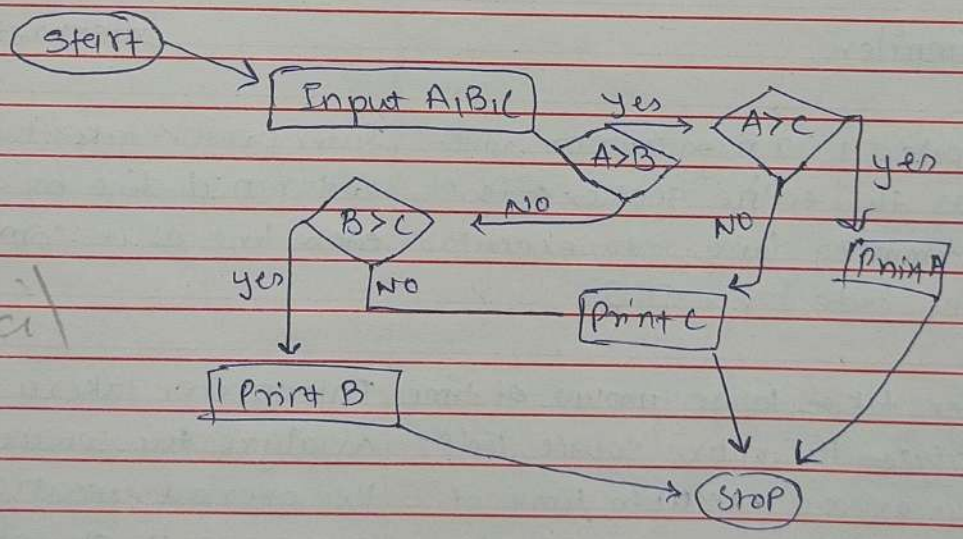
Generate intermediate object code. Example: C, C++, java	No intermediate object code is generated. Example: python, perl.

Q-4) Define algorithm. Write in algorithm to find the greatest number out of 3. Also draw its flowchart.

→ Algorithms are simply a series of instructions that are followed step by step to do something careful and solve a problem.

- The word Algorithm means "a process of set of rules to be followed in calculations or other problem solving operations".

1) Start
2) Input A, B, C
3) If (A>B) and (A>C) then perint "A is greater"
4) Else if (B>A) and (B>C) then print "B is greater"
5) Else print "C is greater"
6) Stop.



Use pencil

Q5 Compare between python 2 and python 3. Write down the feature of python programing language.

python - 2	python - 3.
Syntax is comparatively difficult to understand	syntax is simpler and easily understandable.
ASCII strings typer is used by difficult default to store things.	Unicode is the implicit string typer by default.
The print syntax is treated as statement rather than a function which requires text to wrapped in paranthesis print" Hello word"	The print is explicity treated is a function and replaced by Print() function in python 3 which requires the extra pair of paranthesis
It simply returns an integer value to the nearest whole number when dividing two integer	It makes integer division more intuitive by using the division for integers and floats.
In python 2, the ~~range~~ xrange() is used for iterations.	The new Range() functions introduced to perform ;
Many recents developers are creating libraries which you can only use with python3	Many older libraries created for python 2 is not forward - Compatible.

→ Feature of Python:-

1] Easy to Code: Python is a very developer friendly language which mean that anyone and everyone can learn to code it in a couple of hours or days. As compared to other objects

oriented programming language like Java, C++ and C#, python is one of the easiest to learn.

2] Open Source and free: Python is an open source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this python is free to download and use in any operating system, be it Windows Mac or Linux.

3] Support for GUI: GUI or Graphical User Interface is one of the key aspects of any programming language because it has the ability to add flair to code and make the results more visual. Python has support for a wide array of GUIs which can easily be imported to the interpreter, thus making this one of the most favourite language for developers.

4] Object-oriented Approach: One of the key aspects of Python is its object-oriented Approach. This basically means that python recognizes the concept of class and object encapsulation thus allowing programs to be efficient in the long run.

5] High-Level language: Python has been designed to be a high-level programming language which mean that when you code in python you don't need to be aware of the coding structure, architecture as well as memory management.

6] Integrated by Nature: Python is an integrated language by nature. This means that the python interpreter executes codes one line at a time. Unlike other object-oriented programming language we don't need to compile Python code thus making the debugging process with much easier and efficient. Another advantage of this is that upon execution the python code is immediately converted into an intermediate form also known as byte code which makes it easier to execute and also saves runtime in the long run.

7] Highly Portable: Suppose you are running python on Windows and you need to shift the same to either a Mac or a linux system then you can easily achieve the same in Python without having to worry about changing the code. This is not possible in other other programming languages, thus making Python one of the most portable languages available in the industry.

8] Highly Dynamic: As mentioned in an earlier paragraph, Python is one of the most dynamic languages available in the industry today. What this basically mean is that the type of a variable is decided at the run time and not in advance. Due to the presence of this feature we do not need to specify the type of the variable during coding, thus saving time and increasing efficiency.

9] Extensive Array of Library: Out of the box, Python comes inbuilt with a large number of libraries also make sure that you don't need to write all the code yourself and can import the same from those that already exist in the libraries.

10] Support for Other languages: Being coded in C, python by default supports the execution of code written in other programming languages such as Java, C and C++ thus making it one of the versatile in the industry.

7]. Explain the operator in python.

→ 1) Arithmetic Operation

Arithmetic operator are used to perform mathematical operations like addition, subtraction, multiplication, etc.

10/01/2023 19:19

Operator	Ex	Operator	Ex.
+	x+y+z	/	x/y
-	x-y-z	% Modulus of division left operand by right	x%y
x	x*y	**	
		// Floor division	
		** Exponent left operator raised to the power of right	

Operator	Description	Syntax.
and	Logical And True if both the operands are true	x and y
or	Logical or : True if either of the operands is true.	x or y
not	Logical Not :- True if operand is false	not x.

Bitwise Operator:-

Operator	Description	Syntax.
&	Bitwise AND	x & y
!	Bitwise OR	x | y
~	Bitwise NOT	~x.
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x >> y
<<	Bitwise left shift	x << y

→ Membership Operator

Python membership operator are used the membership of value inside a python data structure. If the value is present in the data structure, then the resulting value is true otherwise it return false.

Operators	Meaning	Example
in	True if value/variable is found in the sequence	sin x
not in	True if value/variable not found in the sequence	5 not in x

- Identity operators.

Is and is not are the identity operators in python. They are used to check if two values are located on the same part of the memory. two variables that are equal does not imply that they are identical.

Operator	Ex	Meaning:
Is	x is true	True if the operands are identical (refer to the same objects).
is not	x is not true	True if the operands are not identical (do not refer to the same object).

Q-8] Describe the Strings in python. Explain any 10 strings manipulation functions with example.

- A string is a sequence of character

- A character is simply a symbol for Example the English lang. has 26 character.

- Computer do not deal with character, the deal with numbers. Even though you may see characters on your screen, internally it is stored and manipulated as a combination of 0s and 1s

	Description
1) Capitalize ()	Converts the first to upper case

Print ("capitalize ==", str. capitalize ())

2) Center ()	Returns a centered string

Print ("center ==", str. center (40))

3] Startwith ()

Returns the number & times a specified value occurs in a string

Print ("startswith he ==", str. start with ("he")).

4] endwith ()

Returns true if the string ends with the specified value

Print ("endswith lo ==", str. endswith ("lo")).

5] find ()

Searches the string for a specified value and return the position of where it was found.

Print ("find ==", str. Find ("el")).

6] isspace ()

Returns True if all character in the string are whitespacer

Print ("isspace ==", str. isspace ()).

7] istitle ()

Returns true if the string follows the rules of a title

Print (" istitle ==", str. istitle ()).

8) join ()

joins the elements of an iterable to the end of the string.

Print (" join ==", '—'. join (str)).

9] isupper ()

Returns True if all characters in the string are upper case.

Print ("isupper ==", str. isupper ())

10] Split()

Split the string at the specified separator and returns a list

Print (" split ==', str. split ()).

10/01/2023 19:19

Python Assignment 2.

INSTITUTE OF TECHNOLOGY & MANAGEMENT UNIVERSE, VADODARA.

ITM UNIVERSE

DATE: / /
CLASS:

Sheet No. 01

Q1 Explain the datatype List and its inbuilt Methods.

→ Lists :-

* Lists are used to store multiple items in a single variable.

→ Lists are one of 4 built-in data types in python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

→ A list is a container which holds comma-seperated values (items or elements) between square brackets.

→ Lists need not to be homogeneous always which makes it a most powerful tool in python.

→ A single list may contain Data type like, Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

→ Lists in python are ordered and have finite definite count.

→ The elements in a list are indexed according to a definite sequence and the indexing of a list is done with 0 being a fixed first index.

→ Each element in the list has its definite place in the list, which allows duplicating of element in the list, with each element having its own distinct place and credibility.

* Built-In Methods:-

→ While a function is what you can apply on a construct and get a result, a method is what you can do to it and change it. To call a method on a construct, you use the dot-operator (.) ~~an~~ python supports some built-in methods to alter an python list

10/01/2023 19:20

	Name	Description	Example
1.	append ( )	It adds an item to the end of list.	a=[1,2,3] a.append(4).
2.	insert ( )	It inserts an item at a specified position.	a. insert(3,5).
3.	remove ( )	It removes the first instance of an item, from the python list	a=[2,1,3,5,4] a.remove(2).
4.	Pop ( )	It removes the element at the specified index, and prints it to the screen.	a.pop(3)
5.	clear( )	It empties the python list	a.clear( )
6.	index ( )	It returns the first matching index of the item specified.	a=[1,3,5,3,4] a.index(3).
7.	count( )	It returns the count of the item specified.	a.count(3).
8.	sort ( )	It sorts the list in an ascending ~~offer.~~ order.	a.sort( )
9.	reverse( )	It reverses the order of elements in the python lists.	a.reverse( )

Q.2 Explain the datatype tuple and its inbuit methods:-

→ Tuple:-

- Tuples are used to store multiple items in a single variable.

→ A tuple is a collection which is ordered and unchange able ie. immutable.

→ Tuples are written with round brackets.

10/01/2023 19:20

- Tuples are sequences, just like lists. The difference between tuples and list are the tuples cannot be changed unlike lists and tuples use paranthesis, whereas lists use square brackets.

- Tuples can have any number of items and they may be of different types (integer, float, list, string, etc...).

- Creating a tuple is as simple as putting different comma-seperated values. Optionally you can put these comma-seperated values between paranthesis also.

→ Tuple Inbuit-methods :-

- Methods that add items or remove items are not available with tuple. Only the following two methods are available

- Some example of methods of python tuple.

- my-tuple = ('a', 'p', 'p', 'l', 'e',)
- print (my_tuple).count ('p') → 2.
- print (my_tuple).index ('l') → 3

Q.3 Explain the datatype set and its inbuit methods.

→ Sets :-

→ Sets are used to store multiple items in a single variable.

- A set is a unordered collection of items. Every set element is unique (no duplication) and must be immutable (cannot be changed).

- However, a set itself is mutable, we can add or remove items from it.

10/01/2023 19:20

→ Sets can also be used to perform mathematical set operation like union, intersection, symmetric difference, etc.

→ In-built Methods in Set:

	Methods	Description.
1.	add ( )	Add an element to the set
2.	clear( )	Removes all element from the set
3.	copy ( )	Returns the ~~difference~~ a copy of the set.
4.	difference ( )	Returns the difference of two or more sets as a new set
5.	difference_update ( ).	Removes all elements of another set from this set.
6.	discard( )	Removes an element from the set if it is a member.
7.	intersection( )	Returns the intersection of two sets as a new set
8.	is disjoint( )	Returns true if two sets have null Intersection.
9.	intersection_update( )	updates the set with the intersection of itself and another
10.	issubset( )	Returns True if another set contains this set
11.	issuperset ( )	Returns True if this set contains another set
12.	POP( )	Removes and returns an arbitrary set element. Raises keyError if the set is empty.
13.	remove( )	Removes an Element from the set. If the element is not a member raises a keyError.
14.	symmetric_difference( )	Returns the symmetric difference of two sets as a new set
15.	Symmetric_difference_update( )	Updates a set with symmetric differences of itself and another.
16.	union( )	Returns the union of sets in a new set.
17.	update( )	Updates the set with the union of itself and others.

10/01/2023 19:21

Q.4 Explain the datatype dictionary and its inbuilt methods.

→ Dictionary:-

→ Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair.

→ Dictionary is a mutable data type in Python.

→ A python dictionary is a collection of key and value pairs seperated by a colon(:), enclosed by in curly braces.

→ Here we have a dictionary, left side of the colon(:) is the key and right side of the(:) is the value.

mydict = {'Name': 'Tanvi', 'Age': '27', 'City': 'Vadodara'}

→ Dictionaries are optimized to retrieve values when the key is known.

→ While the values can be of any Data type and can repeat, keys must be of immutable type (string, number or tuple with immutable elements) and must be unique.

- Empty dictionary:-
  my_dict = {}.

- Dictionary with integer keys
  my_dict = {1: 'apple', 2: 'ball'}.

Dictionary with mixed keys
  my_dict = {'name': 'Tanvi', 1: [2,4,3]}.

10/01/2023 19:21

- We can also create a dictionary using the built-in dict() function
  dict({1: 'apple', 2: 'ball'}).

- From sequence having each item as a pair
  my_dict = dict([(1, 'apple'), (2, 'ball')]).

* **In-Built Dictionary Method:-**

	Method	Description.
1.	Clear()	Removes all items from the dictionary.
2.	copy()	Returns a shallow copy of the dictionary.
3.	fromkeys(seq [v])	Returns a new dictionary with keys from seq and value to v. (default to None).
4.	get(key [,d])	Returns the value of the key. If the key does not exist, returns d. (default to None).
5.	keys()	Returns a new object of the dictionary's keys.
6.	pop(key [,d])	Removes the item with the key and returns its values and key or d if key is not found. If d is not provided and the key is not found, it raises KeyError.
7.	popitem()	Removes and returns an arbitrary item (key, value) Raises KeyError if the dictionary is empty.
8.	setdefault (key [,d])	Returns the corresponding value if the key is in the dictionary. If not, inserts the key with a value of and returns d (default to None).
9.	update([other])	Updates the dictionary with the key/value pairs from other, overwriting existing keys.
10.	values()	Returns a new objects of the dictionary's values.

10/01/2023 19:21

Q5 Differentiate the following with retrence to mutable-immutable object.
a. list and tuple
b. Set and frozen set

a. List and Tuple.

	List	Tuple
1.	Lists are mutable	Tuples are immutable.
2.	The implication of iteration is Time-consuming.	The implication of iterations is comparatively faster.
3.	The list is better for performing operations, such as insertion and deletion.	Tuple data type is appropriate for accessing the element.
4.	List consume more memory.	Tuple consume less memory as compared to the list.
5.	Lists have severeal built-in methods	Tuple does not have so many built-in methods.
6.	The unexpected changes and errors are more likely to occur.	In tuple, it is hard to take place.

10/01/2023 19:21

b. Set and frozen set.

Sets:	Frozensets:
→ Set are used to store multiple items in a single variable.	Frozenset is a new class that has the characteristics of a set.
→ A set is an unordered collection of items. Every set element is unique and must be immutable.	Frozen set are immutable sets. but its element cannot be changed once assigned.
→ How ever, a set is itself is mutable. We can add or remove items from it	Being immutable, it does not have methods that add or remove element
→ Set can also be used to perform mathematical set operations like union, intersection, symmetric diff etc.	Frozenset supports methods of copy(), difference(), intersection(), is disjoint(), issubset(), issuperset() symmetric_difference() and union().

Q.6 Explain the use of range function using an example.

→ The python range() function returns a sequence of numbers, in a given range. The most common use of it is to iterate sequence on a sequence of numbers using python loops.

Syntax: range(start, stop, step)
parameter: Start: [optional] Start value of the sequence.
Stop: next value after the end value of the sequence
Step: [optional] integer value, denoting the difference between any two number in the sequence.
Return: Returns a range type object.

→ In simple terms, range() allows the user to generate a series of numbers within a given range. Depending on how many arguments the user is passing to the function, the user can.

10/01/2023 19:21

decide where that series of numbers will begin and end as well as how big the difference will be between one number and the next.
python range() function takes can be initialized in 3 ways.

- range(stop) takes one argument.

```
eg:- for i in range(6)
        print(i, end=" ")
     print()
```

Output: 0 1 2 3 4 5.

- range (start, stop) takes two arguments.

```
eg:- for i in range(5, 20):
        print (i, end=" ")
        print (i).
```

Output: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19.

- range(start, stop, step) takes three arguments.

```
eg: for i in range (0, 10, 2):
        print (i, end=" ")
        print().
```

Output: 0, 2, 6, 8.

* Some Imp points to remember about the python range function:

range function only works with the integer, i.e. whole numbers.
All the arguments must be integer. User can not pass a string or float number or any other type in a start, stop and step argument of a range().
All arguments can be positive or negative, and Step value must not be zero. If a step is zero then Error occur.
range() is a type in python.
Users can access item in a range() by index, just as ~~order~~ user do with a list.

10/01/2023 19:21

Q7 Explain nested if-else constructs giving a suitable example:

→ All programms use one or more of these constructs. The longer and more complex the program, the more these constructs will be used repeatedly.

• Decision Making in Python:

→ Decision making is required when we want to execute a code only if a certain condition is satisfied.

→ The if.. elif.. else statement is used in python for decision making.

• Python if Statement Syntax:

→ if test expression:
    statements

→ Here, the program evaluates the test expression and will execute statement(s) only if the test expression is True.

→ If the Test expression is false, the statement(s) is not executed.

→ In python the body of the if statement is indicated by the indentation. The body starts with an indentation and the first unindented line marks the end.

→ python interprets non-zero values as True. None and 0 are interpreted as False.

→ python if Statement Flowchart

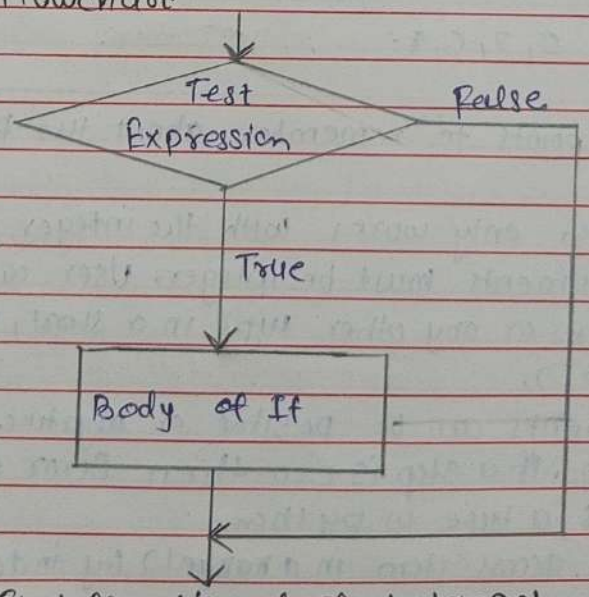


Fig: Operation of If statement

10/01/2023 19:21

Example: of python if-statement:

If the number is positive, we print an appropriate message.

```
num = 3
if num > 0
        print (num, "is a positive number.")
print ("This is always printed.").
```

```
num = -1
if num > 0:
    print (num, "is a positive number,")
print ("This is also always printed)
```

- Python If... else Statement
- Syntax of if...else.

```
if test expression:
        Body of if
else:
        Body of else
```

- The if..else statement evaluates test expression and will execute the body of if only when the test condition is True.
- If the Condition is False, the body of else is executed. Indentation is used to seperate the blocks.
- python if..else Flowchart:

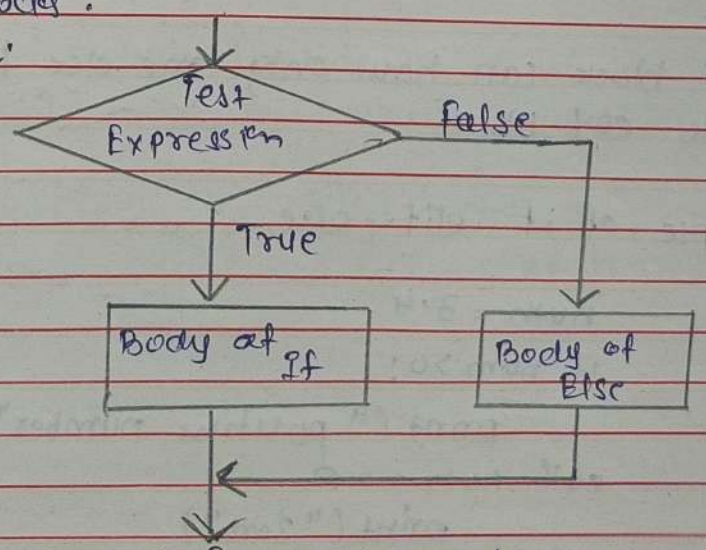


fig: Operation of if...else statement.

10/01/2023 19:21

Example of if... else
program checks if the number is positive or negative

```
num = 3
if num>0
    print ("Positive or zero")
else:
    print("Negative number").
```

- Python if... elif... else Statement
- Syntax of if... elif... else

```
if test expression
    Body of if
elif test expression
    Body of elif
else:
    Body of else.
```

→ The elif is short form for elseif. It allows us to check for multiple expressions.
→ If the condition for if is False, it checks the condition of the Next elif block and so on.
→ If all the conditions are False, the body of else is executed.
→
→ Only one block among the several if... elif... else blocks is executed according to the condition.

- Then it block can have only one else block. But it can have multiple elif blocks.

Example of if... elif... else...

```
num = 3.4
if num >0:
    print (" positive number")
elif num == 0
    print ("zero")
else:
    print ("Negative number")
```

10/01/2023 19:21

* Flowchart of if...elif...else

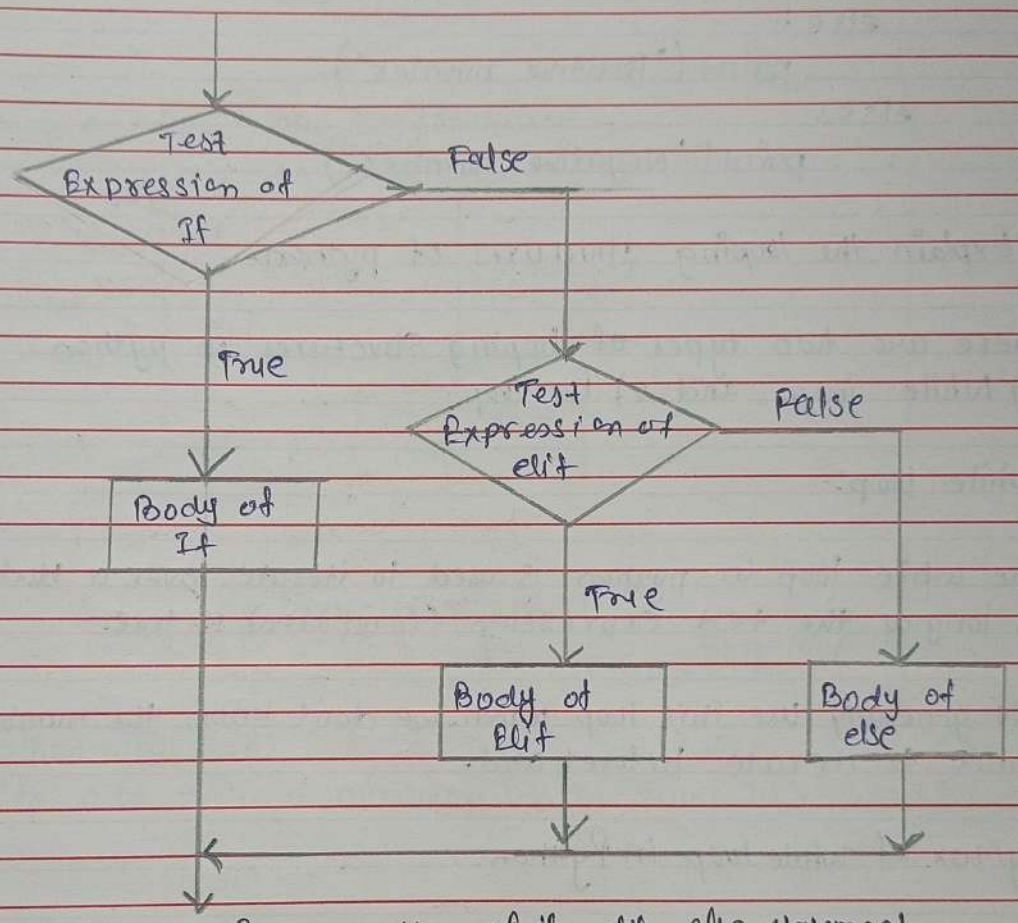


Fig: Operation of if...elif...else statement

- Python Nested if statements
- We can have if...elif...else statement inside another if..elif...else statement. This called nesting ~~and~~ in computer programming.
- Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting.

10/01/2023 19:21

Python Nested if Example.

```
num = float(input("Enter a number:"))
if num>0
    if num ==0:
        print("zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

Q8 Explain the looping structures of python.

→ There are two types of looping structures in python
(i) While loop and 2) for loop.

(i) → While loop:-

→ The while loop in python is used to iterate over a block of code as long as the test expression (condition) is true.

→ We generally use this loop when we don't know the number of times to iterate beforehand.

→ Syntax of while loop in Python.

→
```
while test_expression
    Body of while
```

→ In the while loop, test expression is checked first. The body of the loop is entered only if the test_expression evaluates to True. After one iteration, the test expression is checked again. This process continues until the test_expression evaluates to false.

→ In python, the body of the while loop is determined through indentation

→ The body starts with indentation and the first unindented line marks the end.

→ Python interprets any non-zero value as True, None and 0 are interpreted as false.

10/01/2023 19:21

* Flowchart of while loop:-

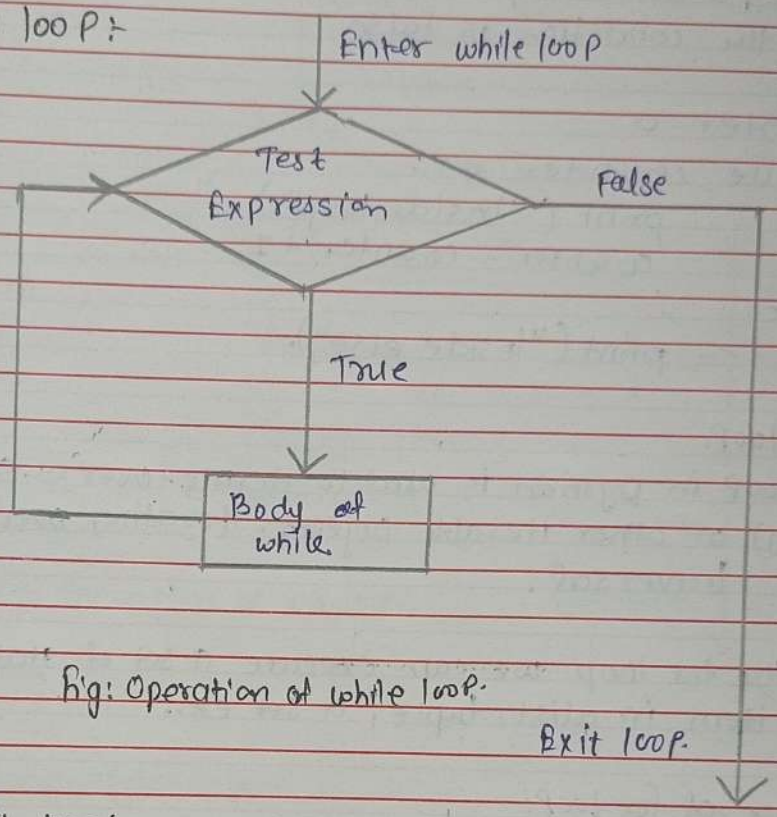


Fig: Operation of while loop.

Example: python while loop:

→ program to add natural numbers up to sum = 1+2+3+...+n.

```
n = 10
sum = 0
i = 1
while i <= n:
    sum = sum + i
    i = i+1
print("The sum is", sum).
```

While loop with else:

While loops can also have an optional else block.

The else part is executed if the condition in the while loop evaluates to False.

10/01/2023 19:21

→ The while loop can be terminated with a break statement. In such cases, the else part is ignored.
→ Hence, a while loop's else part runs if no break occurs and the condition is false

```
counter=0
while counter <3:
        print ("Inside loop")
        counter = counter +1
else:
        print ("Inside else").
```

(ii) for loop:
→ for loop in python is used to iterate over a sequence (list, tuple, string) or other iterable objects. Iterating over a sequence is called traversal.

→ With the for loop we can execute a set of statements, once for each item in a list, tuple, a set etc.

• Syntax of for loop:-

→ for val in sequence
        Body of for

→ Here, val is the variable that takes the value of the item inside the sequence on each iteration.

→ Loop continues until we reach the last item in the sequence, The body of for loop is seperated from the rest of the code using indentation.

→ Example: Program to find the sum of all numbers stored in a list

```
number = [0,5,3,4,2,5,4,1]. → # variable to store the sum
sum =0 → # iterate over the list
for val in numbers:
        sum= sum + val
        print (" The sum is", sum)
```

output:- 0,1,1,2,3,5,8.

10/01/2023 19:21

Flowchart of for loop.

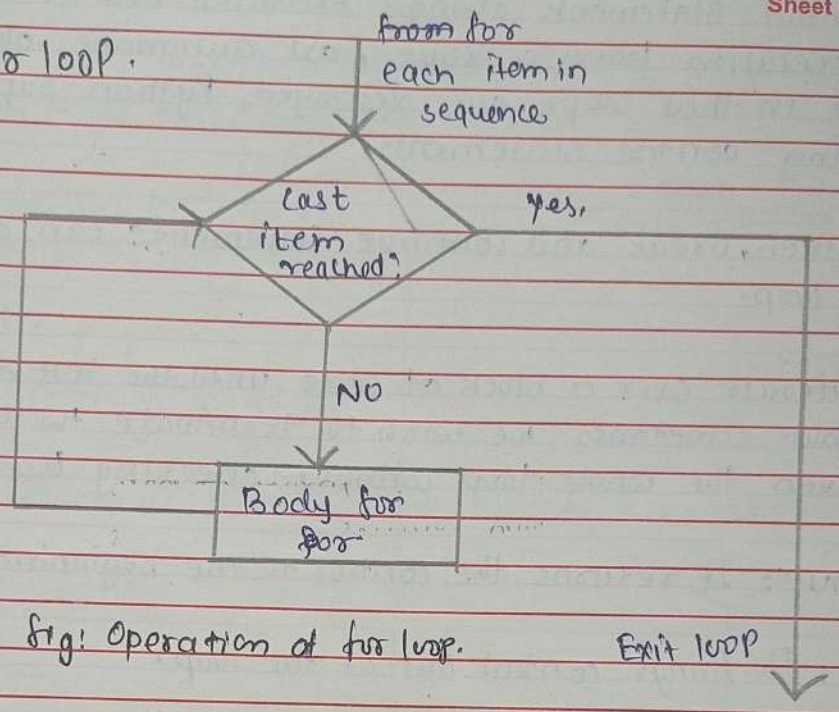


fig: Operation of for loop.

* Nested loops:-

→ A nested loop is a loop inside a loop.

→ The "inner loop" will be executed one time for each iteration of the "outer loop".

eg:-

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]
for x in adj:
  for y in fruits:
    print (x,y)
```

Else in for loop:

The else keyword in a for loop specifies a block of code to be executed when the loop is finished.

eg:- print all no. from 0 to 5, and print a message when the loop ended.

```
for x in range(6):
    print(x)
else:
  print ("Finally finished")
```

10/01/2023 19:21

Q9 Explain break, continue and pass statements of python.

→ Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed, Python supports the following control statements-

→ In python, break and continue statements can alter the flow of normal loop.

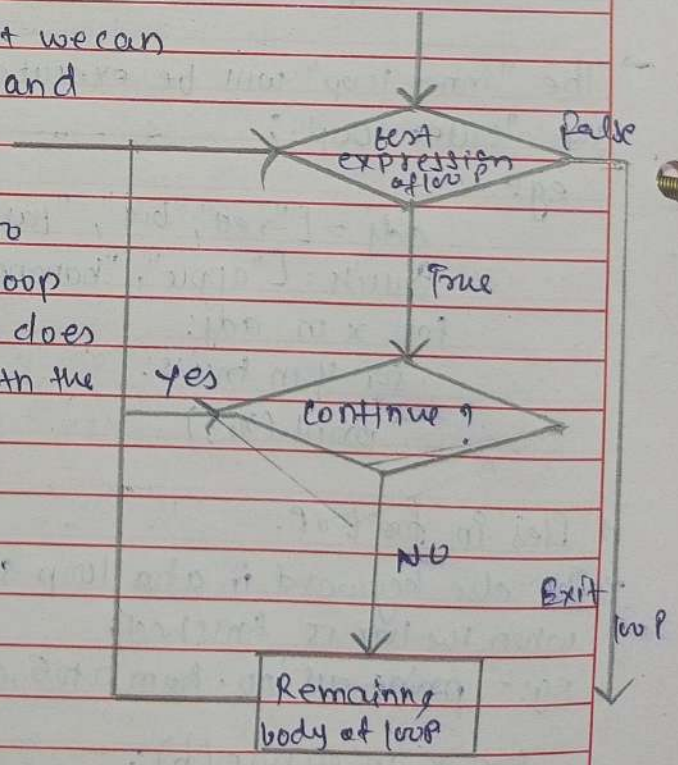
→ Loops iterate over a block of code until the test expression is false, but sometimes we wish to terminate the current iteration or even the whole loop without checking test expression.

→ Continue:- It returns the control to the beginning of the loop.

→ Break: It brings control out of the loop.

→ Pass: we use pass statement to write empty loops. Pass is also used for empty control statement, functions and classes.

* Continue: with the continue statement we can stop the current iteration and continue the next



→ The continuation statement is used to skip the rest of the code inside a loop for the current iteration only. loop does not terminate but continues on with the next iteration.

Example:
```
i=0
while i<6 :
    i+=1
    if i==3:
        continue
    print(i)
```

10/01/2023 19:22