❖ Features of Python: Total Marks - 08

Python provides lots of features that are listed below.

1. **Easy to learn and Use:**
   - Python is very easy to learn language as compared to other language like c, c#, java script, java etc.
   - It is very easy to code in python language.
   - It is developer-friendly language.
   - Python is high level programming language.

2. **Interpreted Language:**
   - Python is an interpreted language i.e. interpreter executes the code line by line at a time.
   - This makes debugging easy.
   - Python code is processed at runtime by the interpreter.
   - You do not need to compile your program before executing it. This is similar to PERL and PHP language.
   - Interpreter generate the code which run on any platform that's via Python is known as Platform Independent programming language.

3. **Free and Open Source:**
   - Python language is freely available at official website and you can download it without purchasing any license copy.
   - It is open source programing language, it means, its source code is also available to the public.

4. **Object Oriented Programming:**
   - Python supports object oriented language and concepts of classes , objects, inheritance,etc.
   - This is one of the key feture of Python programming language.
   - In this programming language, we are focusing on data rather than procedure.
   - Here, we will provide the more security for the data.

5. **Cross-platform and Portable language:**
   - Python language is portable programming language.
   - It means, you can easily move python code from one machine to another machine.
   - For example, if we have code of windows machine then we want to run this code on other machine such as Linux, Unix, Mac OS then we can easily run it without changing any code.
   - We can run Python code on any platform.

6. **Large Standard Library:**
   - Python has a large standard libraries.
   - It provides rich set of modules and functions which help us to develop the Python program.
   - There are many libraries present in python such as regular expressions, unit-testing, web browsers etc.

7. **Dynamically Typed Language:**
   - Python is dynamically-typed language.
   - It means the type (for example- int, float, long etc) for a variable is decided at run time.
   - Because of this feature we don't need to specify the data type of variable.

8. **Integrated Language:**
   - Python is an Integrated programming language because we can easily integrate python with other language like c, c++ etc.

9. **High-Level Language:**
   - Python is a high-level language.
   - When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

10. **GUI Programming Support:**
    - Graphical Users interfaces(GUI) can be developed using a python module such as PyQt5, PyQt4, wxPython or Tk in python.
    - PyQt5 is the most popular option for creating graphical apps with Python.

11. **Support Garbage Collection:**
    - Python deletes unwanted objects (built-in types or class instances) automatically to free the memory space.
    - The process by which Python periodically frees and reclaims blocks of memory that no longer are in use is called Garbage Collection.

❖ History of Python:

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in the December 1989 by Guido Van Rossum at Centrum Wiskunde & Informatica (CWI) in Netherland.
- In 1994, Python 1.0 was released with features like: lambda, map, filter, and reduce.
- Python 2.0 added new features like: list comprehensions, garbage collection system.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.

❖ Applications of Python:

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science.

1) Web Applications.

2) Desktop GUI Applications

3) Software Development

4) Scientific and Numeric

5) Business Applications

6) Console Based Application

7) Audio or Video based Applications

8) 3D CAD Applications

9) Enterprise Applications

10) Applications for Images

Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc.

❖ Python Keywords:

- Keywords are the reserved words in Python.
- We cannot use keyword as a variable name, function name or some other purpose.
- Each and every keywords has some special meaning.
- In Python, keywords are case sensitive.
- All the keywords should be written in lower case except True, False and None.
- The list of all the keywords are given below. This list can vary slightly in the course of time.

Keywords in Python

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

❖ Python Identifiers:

- An identifier is a name which given to class, functions, variables, etc.
- It helps to differentiate one entity from another.
- All user defined name which we use in Python code is known as Identifier. We use name for the identification purpose.
- **Following are the rules which we use while constructing the Identifier Name.**

1. Identifiers name must start with a alphabets or the underscore(_) character.
2. It may combine with Numbers(0-9).
3. Identifiers name should not start with Numbers(0-9).
4. Keywords cannot be used as an Identifier name.
5. It does not allowed any white space between Identifiers name.
6. It does not allowed any special symbols except underscore in Identifiers name.
7. Identifiers names are case-sensitive (Eg. VJTECH, VJTech and vjtech are three different variables).
8. Identifier name can be of any length.

Example:

```
#Valid identifier names:
vjtech
vj_tech
_vj_tech
vjTech
VJTECH
vjtech123

# Invalid identifier names:
2vjtech          #begin with number 2
vj@tech          #special symbol @ present
VJ Tech          #white space used
```

❖ Python Variables:

- Variable is an Identifier which can change value during the execution of program.
- Variable act as a container for storing the value.
- A variable is a named location used to store data in the memory.
- In python, there is no need of declaration of variables by using the data type.
- We can directly use the variable name for storing the value and its data type is decided dynamically as per the stored value on it.
- **Following are the rules which we use while constructing the variable Name.**

1. Variable name must start with a alphabets or the underscore(_) character.
2. It may combine with Numbers(0-9).
3. Variable name should not start with Numbers(0-9).
4. Keywords cannot be used as an Variable name.
5. It does not allowed any white space between Variable name.
6. It does not allowed any special symbols except underscore in Variable name.
7. Variable name is case-sensitive (Eg. VJTECH, VJTech and vjtech are three different variables).
8. Variable name can be of any length.

Example:

```
#Valid variable names:
vjtech = "Vishal Sir"
vj_tech = 123
_vj_tech = 34.56
vjTech = 3+5.6j
VJTECH = (10,20,30,40)
vjtech123= [12,4.5,67,89]

# Invalid variable names:
2vjtech = 123
vj#tech = "Vishal Sir"
VJ Tech = 678.99
```

- **Assigning a value to a Variable in Python:** you can use the assignment operator (=) to assign a value to a variable.
- Example 1: Declaring and assigning a value to a variable

```
Roll_No = 1010;
print("My Roll No:",Roll_No);
```

When you run the program, the output will be:

```
My Roll No:1010
```

- Example 2: Changing the value of variable

```
Roll_No = 1010;
print("My Roll No:",Roll_No);
Roll_No = 2020;
print("My Roll No:",Roll_No);
```

When you run the program, the output will be:

```
My Roll No:1010
My Roll No:2020
```

- Example 3: Assigning multiple values to multiple variables

```
a, b, c = 56, 3.2, "Hello";

print("Value of a=",a);
print("Value of b=",b);
print("Value of c=",c);
```

When you run the program, the output will be:

```
Value of a=56
Value of a=3.2
Value of a=Hello
```

- Example 4: If we want to assign same value to multiple variables.

```
a=b=c="Hello";
print("Value of a=",a);
print("Value of b=",b);
print("Value of c=",c);
```

❖ Python Indentation:

- Most of the programming languages like C, C++, Java use curly brackets { } to define a block of code. Python uses indentation.
- A code block (body of a function, loop etc.) starts with indentation and ends with the first unindented line.
- The amount of indentation is up to you, but it must be consistent throughout that block.
- If we use indentation in Python then the code look neat and clean.
- Generally four whitespaces are used for indentation or we can use single Tab.
- **Example:**

```
for i in range(1,11):
    if i == 5:
        break
    print(i)
```

❖ Python Comments:

- Comments are very important while writing a program.
- It describes what's going on inside a program so that a person looking at the source code can understand the code easily.
- You might forget the key details of the program you just wrote in a mon th's time. So taking time to explain these concepts in form of comments is always helpful.
- Comment is a part of documentation which helps us to give more detail information about the code.
- Python Interpreter ignores the comments, it will not run.
- There are two types of comments present in Python.

  1. Single line comments
  2. Multi-line comments

Single line comments:

- In Python, we use the hash (#) symbol to write a single line comment.
- Single line comment begin with # symbol and end with the end of line.
- We can cover multiple lines by using single line comment, for that you have to write # symbol for beginning of each line.
- Example:

```
#This is Python Program which display Message.
#This program developed by VJTech Academy

print("Welcome to world of Python Language");
```

Multi-line Comments:

- In Python, multi-line comments begin with three times either single quotes (''') or double quotes (""") and end with three times either single quotes (''') or double quotes (""").
- These triple quotes are generally used for multi-line strings.
- But they can be used as multi-line comment as well.
- Example:

```
"""This is Python Program which display Hellp Message.
This program developed by VJTech Academy"""

print("Welcome to world of Python Language");
```

❖ Data Types in Python:

- Every value in Python has a datatype.
- Variables can hold values of different data types.
- Python is a dynamically typed language hence we need not define the data type of the variable while declaring it.
- The interpreter implicitly binds the value with its data type.
- We can check the data type of the variable used in the program by using predefined function **type()** which returns the data type of the variable.
- Consider the following example to define the values of different data types and checking its type.

```
a=10
b=12.45
c="VJTech"
print("Value of a=",a," and data type is",type(a));
print("Value of b=",b," and data type is",type(b));
print("Value of c=",c," and data type is",type(c));
```

When you run the above program, the output will be:

```
Value of a=10 and data type is <type, 'int'>
Value of b=12.45 and data type is <type, 'float'>
Value of c=VJTech and data type is <type, 'str'>
```

Following are the data types present in Python:

1. Numeric Types (int, float, complex)
2. String Type (str)
3. Sequence Types (list, tuple, range)
4. Mapping Type (dict)
5. Set Types (set, frozenset)
6. Boolean Type (bool)
7. Binary Types (bytes, bytearray, memoryview).

❖ Numeric Data Types:

- Integers, floating point numbers and complex numbers present under the Numeric Data Type category.
- They are defined as **int**, **float** and **complex** class in Python.

Integer (int):

- Integer is a combination of 0 to 9 digits numbers without decimal point.
- Integer number may be positive or negative.
- Integers can be of any length, it is only limited by the memory available.
- Example:

```
a=10
print("Value of a=",a," and data type is",type(a));
```

Floating Point Number (float):

- Floating point number is a combination of 0 to 9 digits number with decimal point.
- Floating point number may be positive or negative.
- A floating point number is accurate up to 15 decimal places.
- Float can also be scientific numbers with an "e" to indicate the power of 10.
- Example:

```
a=10.45
b=12e4
print("Value of a=",a," and data type is",type(a));
print("Value of b=",b," and data type is",type(b));
```

Complex Number (complex):

- Complex number is a combination of real and imaginary parts.
- In python, complex numbers are written with a "j" as the imaginary part.
- Complex numbers are written in the form, x + yj, where x is the real part and y is the imaginary part.
- Example:

```
b=4+3.4j
print("Value of b=",b," and data type is",type(b));
```

❖ Python Strings:

- The collections of characters is known as String. Characters may be alphabets, numbers or special symbols.
- String should be represented by using single quotes or double quotes.
- For String data type,predefined class **str** is used.
- Positive and Negative index associated with the String value.
- String positive index should begin with 0 and end with SIZE-1.
- Negative index should begin with -1 from the last element.
- Multi-line strings can be denoted using triple quotes, ''' or """.
- When we use plus (+) sign with String then it work as concatenation operator and when we use asterisk(*) sign with String then it work as repetition operator.
- Strings are immutable. It means once it is created, you can not change it later.
- Subscript [] and index number is used to access any particular element from the string.
- We can use slice [:] operators to access the data of the strings.
- Example:

```
str1="VJTech Academy";
print(str1);
print(str1[:]);
print(str1[7:]);
print(str1[2:6]);
print(str1+" Awasari");
print(str1*2);
```

When you run the above program, the output will be:

```
VJTech Academy
VJTech Academy
Academy
Tech
VJTech Academy Awasari
VJTech AcademyVJTech Academy
```

❖ Python List:

- List is an ordered sequence of items.
- It is one of the most used datatype in Python and is very flexible.
- All the items in a list do not need to be of the same type.
- List is a collection of mixed data types of items.
- List should be represented by using square bracket [ ].
- For List data type,predefined class **list** is used.
- Positive and Negative index associated with the list items.
- List positive index should begin with 0 and end with SIZE-1.
- Negative index should begin with -1 from the last element.
- Lists are mutable. It means once it is created, you can change it later.

- Subscript [] and index number is used to access any particular element from the list.
- We can use slice [:] operator to access the items of the list.
- Example:

```
a = [5,10,15,20,25,30,35,40]
print(a)
print(a[2])
print(a[0:3])
print(a[5:])
```

When you run the above program, the output will be:

```
[5, 10, 15, 20, 25, 30, 35, 40]
15
[5, 10, 15]
[30, 35, 40]
```

❖ Python Tuple:

- Tuple is an ordered sequence of items.
- All the items in a Tuple do not need to be of the same type.
- Tuple is a collection of mixed data types of items.
- Tuple should be represented by using parentheses ().
- For Tuple data type,predefined class **tuple** is used.
- Positive and Negative index associated with the Tuple items.
- Tuple positive index should begin with 0 and end with SIZE-1.
- Tuple index should begin with -1 from the last element.
- Tuple are immutable. It means once it is created, you can not change it later.
- Subscript [] and index number is used to access any particular element from the Tuple.
- We can use slice [:] operator to access the items of the Tuple
- Tuples are used to read only data and it is faster than list as it cannot change dynamically..
- Example:

```
a = (5,10,15,20,25,30,35,40)
print(a)
print(a[2])
print(a[0:3])
print(a[5:])
```

When you run the above program, the output will be:

```
(5, 10, 15, 20, 25, 30, 35, 40)
15
(5, 10, 15)
(30, 35, 40)
```

❖ Python Set:

- Set is an unordered collection of unique items.
- Set is defined by values separated by comma inside braces { }.
- Items in a set are not ordered.
- We can perform set operations like union, intersection on two sets.
- Set have unique values. They eliminate duplicates.
- Set are unordered collection of items and index numbers are not associated with it. Hence the slicing operator [:] does not work.
- Example:

```
a = {5,10,15,20,25,30,35,40}
print(a)
```

When you run the above program, the output will be:

```
{35, 5, 40, 10, 15, 20, 25, 30}
```

❖ Python Dictionary:

- Dictionary is an ordered collection of items and it should be represented by using (key:value) pairs format.
- It is generally used when we have a huge amount of data.
- Dictionaries are optimized for retrieving data. We must know the key to retrieve the value.
- In Python, dictionaries are defined by using curly bracket {} with each item being a pair in the form key:value.
- Key and value can be of any data type.
- Example:

```
d = {1:'Pune', 2:'Solapur', 3:'Tuljapur', 4:'Thane'};
print("First city is ",d[1]);
print("Fourth city is ",d[4]);
print (d);
print (d.keys());
print (d.values());
```

When you run the above program, the output will be:

```
First city is  Pune
Fourth city is  Thane
{1: 'Pune', 2: 'Solapur', 3: 'Tuljapur', 4: 'Thane'}
dict_keys([1, 2, 3, 4])
dict_values(['Pune', 'Solapur', 'Tuljapur', 'Thane'])
```

❖ Python Type Conversion (Type Casting):

- The process of converting the value of one data type to another data type is called as Type conversion.
- It is also known as Type casting.
- Python promotes conversion of lower datatype (integer) to higher data type (float) to avoid data loss.
- There are two types of type conversion.
  1. Implicit Type Conversion
  2. Explicit Type Conversion

Implicit Type Conversion:

- The type casting which is done by system is known as Implicit type conversion.
- Python automatically convert one type value to another type.
- It does not require any user involvement.
- Example:

```
a = 123
b = 1.23
c = a + b

print("datatype of a:",type(a))
print("datatype of b:",type(b))

print("Value of c:",c)
print("datatype of c:",type(c))
```

When you run the above program, the output will be:

```
datatype of a: <class 'int'>
datatype of b: <class 'float'>
Value of c: 124.23
datatype of c: <class 'float'>
```

Explicit Type Conversion:

- The type casting which is done by programmer is known as Explicit type conversion.
- Here Programmer will convert the data type of an object to required data type.
- We use the predefined functions like int(), float(), str(), etc to perform explicit type conversion..
- Syntax:

```
Variable_Name = (required_datatype) (Expression);
```

- Example:

```
a = 123
b = 1.23
```

```
c = (float)(a + b)

print("datatype of a:",type(a))
print("datatype of b:",type(b))

print("Value of c:",c)
print("datatype of c:",type(c))
```

When you run the above program, the output will be:

```
datatype of a: <class 'int'>
datatype of b: <class 'float'>
Value of c: 124.23
datatype of c: <class 'float'>
```

❖ Python Input and Output Operations:

- There are two built-in functions print() and input() are used to perform I/O operations in Python.

Python Output Using print() function:

- print() function is used to print output messages on output screen.
- This is predefined function of Python language.
- Syntax:

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

Where:

objects -> Value to be printed.

sep -> The separator is used between the values. It's default value is a space character.

end -> After all values are printed then end is printed. It's default value is a new line.

file -> The file is the object where the values are printed and its default value is sys.stdout (screen)

- Example:

```
print(1,2,3,4)
print(1,2,3,4,sep='*')
print(1,2,3,4,sep='#',end='&')
```

When you run the above program, the output will be:

```
1 2 3 4
1*2*3*4
1#2#3#4&
```

Python Input Using input() function:

- input() function is used to take input from user through keyboard.
- This is predefined function of Python language.
- Whatever the value returned by using input() function, by default it is string type value. To convert this into a number, we can use int() or float() functions.
- Syntax:

```
input([message])
```

Where:

message -> This is the string which we wish to display on the screen. It is optional.

- Example:

```
no=input("Enter any Value: ")
print("You have entered value of no= ",no);
```

When you run the above program, the output will be:

```
Enter any Value: 123
You have entered value of no=  123
```