17002010702I

Page No.
Date

Assignment 1

Q.1 Diff. bet^n C and Python

Ans:

C	Python
→ General purpose programming language.	→ General purpose object oriented language.
→ Machine dependent	→ OS dependent.
→ Structured programming language.	→ Object oriented language.
→ Variable declaration necessary.	→ Variable declaration not necessary
→ Compiled by compiler	→ Interpreted by interpreter.
→ Limited built in functions	→ Large library of built in functions

Q.2 Built in datatypes of python. Diff. operators in python. Membership and identity operators with example.

Page No.
Date

Ans ⇒ Built In Data Types:

① Text Type: str → String

② Numeric Types: int → Integer
float → Floating Point Number.
complex → Complex Nos.

③ Sequence Types: list → List
tuple → Tuple
range → Range

④ Mapping Types: dict → Dictionary

⑤ Set Types: set → Set
frozenset → Frozenset

⑥ Boolean Types: bool → boolean

⑦ Binary Types: bytes, bytearray, memoryview

⇒ Operators in Python:

① Arithmetic Operators:
Addition → '+' & x+y

Page No.
Date

Subtraction → '-' x-y
Multiplication → '*' x*y
Division → '/' x/y
Modulus → '%' x%y
Floor division → '//' x//y
Exponentiation → '**' x**y

② Assignment Operations

= → x=5	+= → x+=3
-= → x-=2	*= → x*=2
/= → x/=4	%= → x%=3
//= → x//=3	**= → x**=3
&= → x&=3	|= → x|=3
^= → x^=3	>>= → x>>=3
<<= → x<<=3	

③ Comparision Operators

Equal → ==
Not equal → !=
Greater than → >
Lesser than → <
Greater than or equal to → >=
Lesser than or equal to → <=

④ Logical Operators

and → Returns True if both are true
or → Returns true if one is true
not → Returns false if true.

Page No.
Date

(5) Identity Operators:

is → Returns true if both are same
is not → Returns true if both are not same

(6) Membership Operators:

in → Returns true if x is present in y
not in → Returns true if x is not in y.

(7) Bitwise Operators:

& → AND	~ → NOT
| → OR	<< → Left shift
^ → XOR	>> → Right Shift.

Q.3 Diff. between compiler and interpreter. How does python interpreter work?

Ans.

Compiler	Interpreter
→ Reads code entirely and translates into m/c code	→ Translates program one line at a time.
→ Intermediate code is generated.	→ No intermediate code generated.
→ Slow	→ Fast
→ C, C++, Java.	→ Python, JavaScript

Page No.
Date

Python Interpreter :

The python code we write is compiled into python byte code, which creates file with extension .pyc. The byte code compilation happens internally and hidden from developer.
The .pyc file is executed by virtual machine. The Virtual Machine just a big loop that iterates through your byte code instructions one by one, to carry out their operations.

Q.11 Explain while loop and for loop with example

Ans: **While loop :**

while loop is a loop which is executed till the condition is true.

```
while condition :
    statements.
```

eg:

```
i = 1
while i <= 6 :
    print (i)
    i += 1
```

op:

```
1
2
3
4
5
6
```

Page No.
Date

For loop:

For loop is used for iterating a sequence over and over.

eg:
```
fruits = ['apple', 'banana', 'cherry']
for x in fruits:
    print(x)
```

OP:
```
apple
banana
cherry
```

Q.5 Explain lists along with methods associate with lists. and explain mutablity with respect to Lists.

Ans:

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. Lists are created using square brackets.

```
list1 = ['apple', 'cherry', 'banana']
```

→ List Methods:

Page No.
Date

① Change item value

eg: list1[1] = 'mango'
print (list1)

op: ['apple', 'mango', 'banana']

② Append items

list1.append('orange')
print (list1)

op: ['apple', 'mango', 'banana', 'orange']

③ Remove items.

list1.remove('banana')
print(list1)

op: ['apple', 'mango', 'orange']

list1.pop(2)

op: ['apple', 'mango']

④ Sort list.

eg: list2 = [1, 2, 3, 4, 5, 6, 3, 2, 1]
list2.sort()
print(list2)

op: [1, 1, 2, 2, 3, 3, 4, 5, 6]

Page No.
Date

⑤ Copy a list.

eg: list3 = list1.copy()
print(list3)

op:
['apple', 'mango']

⑥ Join a list

eg: list4 = list1 + list2
print(list4)

op:
['apple', 'mango', 1, 1, 2, 2, 3, 5, 4, 5, 6]

→ Mutability:

It is the property of an object to be changed after they have been created.
Lists are mutable.

Q.] What is dictionary in Python? Explain all built in dictionary methods with example.

Ans:

Dictionary:
Dictionaries are used to store data in key : value pairs.

Page No.
Date

Dictionary is a collection which is ordered, changeable and does not allows duplicates.

Dictionary have curly braces and key value pairs.

```
dict1 = { "name" : "Akshat",
          'sem' : 8 }
```

(1) Clear.

dict1.clear() → Empties the dict.

~~print~~

(2) Add items.

```
dict1 ["color" : "red"]
print(dict1)
```

eg: {'color': 'red'}

(3) Remove items.

dict1.pop(color)

(4) Get by values.

```
dict2 = {'1': 2', 'name': 'Akshat'}
dict2.values()
~~print(dict2.values())~~
```

Page No.
Date

Q.9 What is lambda function in python? Explain with example

Ans: Lambda function is a small anonymous function
A lambda function can take any no. of arguements, but can only have one expression.

Syntax:

lambda arguements : expressions

eg:
```
x = lambda a, b, c : a+b+c
print (x(5, 6, 2))
```

op:

13

Q.12 Explain String functions with an example.

Ans: Strings in python are surrounded either by single quotes on double quotes.

eg:
```
print ("Akshat")
```

op: Akshat

Multiline strings are denoted by """ """.

Page No.
Date

① Length of a string

```
a = "Akshat"
print(len(a))
```

op: 6

② Capitalize.

```
print(a.capitalize())
```

op: Akshat

③ Count.

```
print(a.count('a'))
```

op: 2

④ isalpha()

```
print(a.isalpha())
```

op: True

⑤ Lower.

```
print(a.lower())
```

op: akshat

⑥ Split

```
b = "My name is Akshat"
print(b.split())
```

op: ['My', 'name', 'is', 'Akshat']

1700201070021

Page No.
Date

Q.13 Explain Indexing and Slicing operation for a list in python.

Ans Indexing

Items in a list are given indexes as their positions.

The indexing starts with 0 and ends at the 'n-1', where n is no. of items.

Negative index is used when we have to access the list from the end.

eg:

```
a = ['This', 'is', 'a', 'list']
      0      1     2     3   → Positive indexing
     -4     -3    -2    -1   → Negative indexing
```

Slicing:

A slice is a subset of list elements.

eg:

a[start : stop]

a[0:2] ⇒ ['This', 'is', 'a']

1700201070021

Page No.
Date

Leaving blank any boundary means that go to end of the list. You can also use negative index for slicing.

Q.15 What is duck typing philosophy of Python? Explain diff. types of comments.

Ans: Duck Typing is a concept related to dynamic typing; where the type or class of an object is less important than the methods it defines.
When you use duck typing, you do not check types at all.
Instead, you check presence of a given method or attribute.

Comments:
Can be used to explain python code.
Used to make code more readable
Starts with '#'
Hello
Multiline - comments can be used by using
"""
Hello
Akshat """