Name-Swahil Islam
RegEmail- sarfarazahmed2829@gmail.com
CourseName-DataSciencewithGenerativeAI-Sep'24
Assignment Name - Python
SubmissionDate-13 Oct2024
GitLink-

Ques1

SimpleandEasytoLearn-Python'ssyntaxisclearandreadable
Wedon'tneedtocompilethecodebeforerunningit,makingthedevelopmentprocessfaster and more flexible.
Pythonhasaecosystemofthird-partylibrariesandframeworksthatmakeitversatilefor many domains:
DataScienceandMachineLearning:LibrarieslikeNumPy,pandas,andTensorFlowmake Web Development: Frameworks like Django and Flask are popular for building web applications.
Pythonhasalargeandactiveglobalcommunity

2. **Describetheroleofpredefinedkeywordsin Pythonandprovideexamplesof how they are used in a program.**

Predefined keywords in Python are already defined words in python that have a special meaning.Thesekeywordsarepartofthesyntaxandcannotbeusedasvariablenames, function names or identifiers. They help to define the syntax of Python code.

Somecommonpredefinedkeywordsare:

if,else,elif:Usedforconditionalstatements.
for, while: Used for loops.
True,False:Booleanvalues.
None, Print

3. Compareandcontrastmutableandimmutableobjects in Pythonwithexamples.

Thisconceptmeanswhethertheobjectcanbechangedafterithasbeencreated.

MutableObjects:Theseareobjectswhosevaluescanbechangedaftercreation.Lists, dictionaries, and sets are examples of mutable objects.

For example,
my_list=[1,2,3]
my_list[0]=10

print(my_list)

Output:[10,2,3]

ImmutableObjects:Theseareobjectswhosevaluescannotbechangedaftercreation. Examples are strings, and tuples.

my_string="Hello"
my_string[0] = "h"

Thiswillcauseanerror

1. Youcanappendelementstoalistanditwillstillbethesameobject.
2. Ifyouwanttomodifyanimmutableobject,youhavetocreateanewobjectwiththe desired changes.

Ques4

InPython,operatorsaresymbolsthatperformoperationsonvariablesandvalues.Thereare several types of operators in Python

1. ArithmeticOperator
2. AssignmentOperator
3. LogicalOperator
4. ComparisonOperator
5. BitwiseOperator
6. MembershipOperator
7. Identity Operator

Examples,howtheyareused:

Arithmetic Operator

Wehave,
+
-
*
**-Exponential
/-
//-FloorDivisioneg.5//2=2
% - Remainder

AssignmentOperato

r

=
+=a=a+5
-=
*=
/=
Similarly,wecanapplyallarithmeticoperator

Comparison Operator

>
<
>=
<=
==equalto
!=not equal to

LogicalOperato

r

AND-ReturnsTrueifbothconditionsaretrue
or ReturnsTrueifoneoftheconditionsistrueTrueorFalse=True Not
returns False if the result is true not True = False

BitwiseOperators

Bitwiseoperatorsworkonbitsandperformbit-by-bitoperations.

a=10#1010inbinaryusing(2raisetopowertill3ruletaughtinclass)
b=4#0100inbinary

print(a&b)#0(0000inbinary)
print(a|b)#14(1110inbinary)

MembershipOperators
Thisisusedtotestifanelementisamemberofanydatastructure.

in
ReturnsTrueifavalueisfoundinthesequence Eg -
'a' in 'apple' = True

notin
ReturnsTrueifavalueisnotfoundinthesequence Eg
- 'x' not in 'apple' = True

IdentityOperators

Identityoperatorsareusedtocomparethememorylocationoftwoobjects.

Is
ReturnsTrueifbothvariablespointtothesameobject is
not
ReturnsTrueifbothvariablesdonotpointtothesameobject

```
a=[1,2,3]
b=a
c=[1,2,3]

print(aisb)#True(aandbrefertothesameobject)
print(aisc)#False(aandcrefertodifferentobjectswithsamevalues)
```

Ques4-

Typecastingistheprocessofconvertingonedatatypeintoanother.InPython,typecasting can be implicit (automatically done by Python) or explicit (done manually by the us). This is useful when we need to perform operations that require a specific data type.

Typeofthis-
ImplicitTypeCasting-

```
#Implicitlyconvertinginttofloat
num_int = 5
num_float=2.5

result=num_int+num_float
#Pythonconverts'num_int'tofloatautomatically
print(result)Output: 7.5
print(type(result))#Output:'float
```

ExplicitTypeCasting
Inexplicittypecasting,wemanuallyconvertonedatatypeintoanotherusingfunctionslike int(), float(), str(), etc.

int()-Convertstoaninteger
float()-Convertstoafloating-pointnumber
str() - Converts to a string
list() - Converts to a
listtuple() - Converts to a
tuple set() - Converts to a
setbool()-
Convertstoaboolean

Eg.

```
num_float=7.8
```

num_int=int(num_float)
print(num_int)
Output:7(decimalpartisremoved)

Ques5-

Conditionalstatementsallowustomakedecisionsbasedoncertainconditions.Theycontrol the flow of a program by executing specific code when a condition is met and executing different code when the condition is not met (false).

ifstatemen

t age = 18

ifage>=18:
  print("Youareanadult.")

if-

elsestatement

age = 16

ifage>=18:
  print("Youareanadult.")
else:
  print("Youareaminor."

if-elif-

elsestatement

marks = 85

if marks >= 90:
  print("Grade:A")
elif marks >= 80:
  print("Grade:B")
elif marks >= 70:
  print("Grade:C")
else:
  print("Grade:D")

4. NestedifStatements
Wecanuseanifstatementinsideanotherifstatement.Thisiscallednestedif.

```
age = 20
is_student=True

if age>=18:
    if is_student:
        print("You are an adult and a student.")
    else:
        print("You are an adult but not a student.")
else:
    print("You are a minor.")
```

since age is greater than 18 and is_student is True, it will print "You are an adult and a student.".

Ques6-

Loops are used to repeat a block of code multiple times based on certain conditions. There are two types of loops:

for loop
while loop

1. for Loop
The for loop is used to iterate over a sequence (like a list, tuple, string, or range) and **execute a block of code for each item in the sequence.**

```
numbers=[1,2,3,4,5] for

num in numbers:
    print(num)
```
- *execute this block of code for each item in the sequence. So num=1 then num = 2 so on....*

Usecase:

Iterating over a list or string: When we know the sequence we want to iterate over.
We can also use range() to loop a certain number of times.

2. while Loop
The while loop repeats a block of code **as long as the condition is True.** It is useful when we don't know in advance how many times the loop should run, and the loop depends on some condition.

#Printnumbersfrom1to5-here,wedon'tknowinadvancehowmanytimestheloop should run so we will use while loop,

```
i=1
whilei<=5:
    print(i)
    i+=1
```

LoopControlStatements
ThesestatementshelpcontroltheflowofIoops:

break: Exits the loop early when a certain condition is met.continue:Skipsthecurrentiterationandmovestothenextiteration.

```
fornuminrange(1,6):
    if num == 3:
        break
    print(num)

#Output:1,2
```

```
fornuminrange(1,6):
    if num == 3:
        continu
    e
    print(num)

#Output:1,2,4,5
```

NestedLoops
Wecanplacealoopinsideanotherloop.Thisisusefulwhenworkingwithmulti-dimensional condition

ExampleofNestedforLoop:
Printing stars, table etc

Printa3table

```
Num = 3
Foriinnum:#Outerloop
    forjinrange(11):#Innerloop
        print(f"({i}*{j})", end=" ")
```

print()