Artificial Intelligence

LAB 1

- **Programming Syntax**

```
>>>
...print("Hello World!")
   Hello World!
>>>
```

You can also write the code in notepad and run it on command prompt by giving its address.

```
1.py - E:\university stuff\programs\4.Python\1.py (3.13.2)
File  Edit  Format  Run  Options  Window  Help
print("testing if it can read a notepad file")
x=1
if x>0:
        print("Yes, can read ") #printed successfully
```

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5371]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Muhammad>cd ..

C:\Users>cd ..

C:\>e:

E:\>cd university
The system cannot find the path specified.

E:\>cd university stuff

E:\university stuff>cd programs\cd 4. python
The system cannot find the path specified.

E:\university stuff>cd programs

E:\university stuff\programs>cd 4.python

E:\university stuff\programs\4.Python>python 1.py
  File "E:\university stuff\programs\4.Python\1.py", line 4
        print("Yes, can read ") #printed successfully
    ^
```

- **Comments in python**

```
#this is coment
print ("No comments")
```

```
Python 3.13.2 (tags/v3
AMD64)] on win32
Type "help", "copyrigh
>>>
========== RESTART: E
No comments
>>>
```

As you can see in output “Comments” are not printed just code is executed

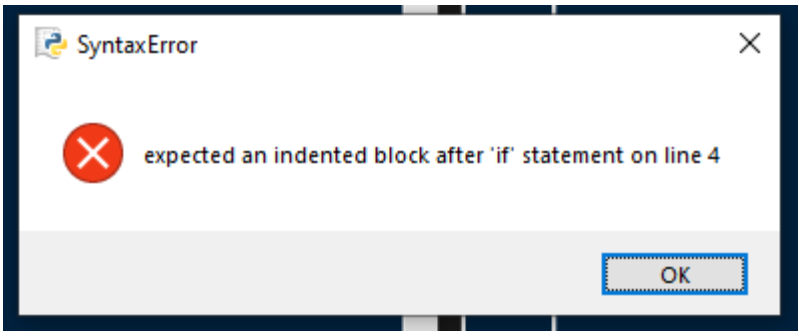
- **Multiple Statements on a Single line**
  To run Multiple statements in python we use; distinguish between two statements

```
print("statement 1")
print("statement 2")
#can also be written as
print("statement 1");print("statement 2")
```

```
>>>
= RESTART: E:\university stuff\programs\4.Python\task 1.py
statement 1
statement 2
statement 1
statement 2
>>>
```

To run Multiple statements in python we don’t have “curly brackets” {}to difference body from conditions.

```
x=1
if x>0:
print("statement 1")
print("statement 2")
```



So, we have to use indentations,

->1 Tab=4 spaces (by default)

```
x=1
if x>0:
    print("statement 1")
    print("statement 2")
```

```
>>>
    = RESTART: E:\university stuff\programs\4.Python\task 1.py
    statement 1
    statement 2
>>>
```

->1 spaces

```
x=1
if x>0:
 print("statement 1")
 print("statement 2")
```

```
    statement 2
>>>
    = RESTART: E:\university stuff\programs\4.Python\task 1.py
    statement 1
    statement 2
>>>
```

->1 Tab and 1 space

```
x=1
if x>0:
	print("statement 1")
	print("statement 2")
```

```
>>>
    = RESTART: E:\university stuff\programs\4.Python\task 1.py
    statement 1
    statement 2
>>>
```

- **Datatypes**

  - **Integer**

```
a=142
print(type(a))

b=-142
print(type(b))

c=0
print(type(c))
```

```
    Python 3.13.2 (tags/v3.1
    AMD64)] on win32
    Type "help", "copyright"
>>>
    =========== RESTART: E:/
    <class 'int'>
    <class 'int'>
    <class 'int'>
>>>
```

  - **Float**

```
a=0.6
print(type(a))

b=-1.5
print(type(b))

c=.45
print(type(c))

d=2.2-1
print(type(d))

e=5E220
print(type(e))
```

```
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.13.2 (tags/v3.13.2:4f8bb
    AMD64)] on win32
    Type "help", "copyright", "credit
>>>
    =========== RESTART: E:/universit
    <class 'float'>
    <class 'float'>
    <class 'float'>
    <class 'float'>
    <class 'float'>
>>>
```

➢ Strings

String can be written between double quotation” “. String can be written between single quotation’ ‘.

```
a="Class"
print(a)

b='Class'
print(b)

e="Day's"
print(e)

f='Day"s'
print(f)
```

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/university stuff/programs/4.Python/;lab 1.py ==========
Class
Class
Day's
Day"s
>>>
```

But starting and ending quotations should be same.

```
a="Class'
print(a)

b='Class"
print(b)
```

SyntaxError

unterminated string literal (detected at line 1)

OK

➢ complex

```
a=complex(1,2)
print(type(a))
print(a)

c=1+2j
print(type(c))
print(c)

d=1+2J
print(type(d))
print(d)
```

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/university stuff/programs/4.Python/;lab 1.py ==========
<class 'complex'>
(1+2j)
<class 'complex'>
(1+2j)
<class 'complex'>
(1+2j)
>>>
```

➢ Boolean

For bool first should be capital.

```
x=True
print(type(x))

y=False
print(type(y))
```

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/university stuff/programs/4.Python/;lab 1.py ==========
<class 'bool'>
<class 'bool'>
>>>
```

- **Escape characters**

```
File  Edit  Format  Run  Options  Window  Help
print("This is a backslash \\ mark.")
print("This is a tab \t mark.")
print("This is a \' single quotation \' mark."
print("This is a  \"double quotation \" mark."
print("This is a new line \nnew line.")
```

```
=========== RESTART: E:\university stuff\programs\4.Python\task 1.py
This is a backslash \ mark.
This is a tab    mark.
This is a ' single quotation ' mark.
This is a  "double quotation " mark.
This is a new line
 new line.
>>>
```

- **Lists**

  - **Numbers**

```
string="Artificial Intelligence "
print(string[0])
print(string[-15])
print(string[14])
print(string[7])
print(string[-11])
print(string[55])
```

```
A
l
e
i
t
```

  - **Strings**

```
color=["red","yellow",'white']
print(color[0])
print(color[0],color[2])
print(color[-1])
print(color[4])
```

we can also display list by using starting and ending point and separated it by colon ":".

task 1.py - E:\university stuff\programs\4.Python\task 1.py (3.13.2)

File Edit Format Run Options Window Help

```
color=["red","yellow","white","blue"]
print(color[0:2])
print(color[1:2])
print(color[1:-2])
print(color[:2])
print(color[:])
```

```
>>>
========== RESTART: E:\university stuff\programs\4.Python\task 1.py ==========
['red', 'yellow']
['yellow']
['yellow']
['red', 'yellow']
['red', 'yellow', 'white', 'blue']
>>>
```

…END…