

Lecture 28

Python



QUIZ

قَالَ رَبِّ اشْرَحْ لِي صَدْرِي ۝
﴿٢٥﴾

[فَالَّذِي نَسِيَ كَهُولَ دَسَّهُ مَنْ يَرَى لَيْسَ بِمَنْ يَرَى
[فَالَّذِي نَسِيَ كَهُولَ دَسَّهُ مَنْ يَرَى لَيْسَ بِمَنْ يَرَى]

وَيَسِّرْ لِي آمْرِي ۝
﴿٢٦﴾

[وَيَسِّرْ لِي آمْرِي : مَنْ يَرَى لَيْسَ بِمَنْ يَرَى]

وَاحْلُلْ عُقْدَةً مِنْ لِسَانِي ۝
﴿٢٧﴾

[وَاحْلُلْ لِي آمْرِي : مَنْ يَرَى لَيْسَ بِمَنْ يَرَى]

يَفْقَهُوا قَوْلِي ۝
﴿٢٨﴾

[يَفْقَهُوا قَوْلِي : مَنْ يَرَى لَيْسَ بِمَنْ يَرَى]

4 QUESTIONS / FEEDBACK / CONCERNS



INFORMATION
TECHNOLOGY
UNIVERSITY

SE SECA SLIDE OF FAME

5



NO ONE
WEEK - 1



Muhammad Daniyal
Hammad (BSSE23046)
WEEK - 2



Syed Hashim Abbas
(BSSE23084)
WEEK - 3



Umar Ahmad
(BSSE23032)
WEEK - 4



Umar Ahmad
(BSSE23032)
WEEK - 5



Fatima Noorulain
BSSE23003
WEEK - 6



Umar Ahmad
(BSSE23032)
WEEK - 7



YOUR NAME
WEEK - 8



YOUR NAME
WEEK - 9



YOUR NAME
WEEK - 10



YOUR NAME
WEEK - 11



YOUR NAME
WEEK - 12



YOUR NAME
WEEK - 13



YOUR NAME
WEEK - 14



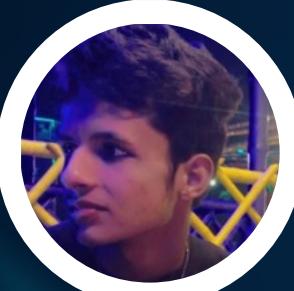
YOUR NAME
WEEK - 15

SE SEC B SLIDE OF FAME

6



Muhammad Mukarram
BSSE23029
WEEK - 1



Muhammad Abdullah
(BSSE23087)
WEEK - 2



Muhammad Abdullah
(BSSE23087)
WEEK - 3



Fasiha Rohail
(BSSE23041)
WEEK - 4



Muhammad Abdullah
(BSSE23087)
WEEK - 5



Hazira Azam
BSSE23019
WEEK - 6



Jamshaid Ahmed
BSSE23012
WEEK - 7



YOUR NAME
WEEK - 8



YOUR NAME
WEEK - 9



YOUR NAME
WEEK - 10



YOUR NAME
WEEK - 11



YOUR NAME
WEEK - 12



YOUR NAME
WEEK - 13



YOUR NAME
WEEK - 14



YOUR NAME
WEEK - 15

RECAP

GitHub

Tools (Cygwin, IDE, GitHub)

Approach towards a word problem

Flowcharts

Flowcharts Advantages & Disadvantages

Algorithms

Pseudocode

Numbers Systems (Decimal, Binary, Octal & Hexadecimal)

Ten's Complement

Twos Complement

main function

Stream in and stream out operators

if else

Functions

Data Types

Arithmetic Operators

Relational Operators

Loops (While, for , do while)

Nested Loops

Switch cases

RECAP

Function Overloading

Scope of variables

Function Prototype and Definition

Default Value in parameters of functions

Parameters by value vs Parameters by Reference

Recursion

Arrays

2D Arrays / Multi Dimensional Arrays

Pointers

Structs

Filing

DMA

Templates

Static Variables

Python

<https://www.programiz.com/python-programming/online-compiler/>

Pycharm

Introduction to Python



INFORMATION
TECHNOLOGY
UNIVERSITY

C++ vs Python

C++

- C++ is a pre-compiled programming language and doesn't need any interpreter during compilation.
- It is a low-level language.
- C++ doesn't support garbage collection.
- C++ is faster in speed as compared to python.

python

- Python is an interpreted language and it runs through an interpreter during compilation.
- It is high-level language.
- Python supports garbage collection.
- Python is slower since it uses interpreter and also determines the data type at run time.



INFORMATION
TECHNOLOGY
UNIVERSITY

Display

C++

```
cout << "hello world";
```

python

```
print("hello world")
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Display

C++
`cout << "hello world";`



python
`print("hello world")`



Comments

C++

```
//This is a comment for single line
```

```
/*
```

This is a comment

written in

more than just one line

```
*/
```

python

```
# This is a comment for single line
```

```
'''
```

This is a comment

written in

more than just one line

```
'''
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Input from user

C++

```
cin >> x;  
  
cout<<"Enter your name:";  
  
cin >> x
```

python

```
x = input()  
  
x = input('Enter your name:')
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Variables

C++

```
int a = 0;  
  
double b = 2.2;  
  
bool flag = True  
  
string name = "noman ali";
```

python

```
a = 0  
  
b = 2.2  
  
flag = True  
  
Name = "noman ali"
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Arrays

C++

```
int cars[3] = [2, 10, 0]
```

```
string cars[3] = ["Ford", "Volvo",
"BMW"]
```

python

```
cars = [2, 10, 0]
```

```
cars = ["Ford", "Volvo", "BMW"]
```

What else can you do in python

```
list= ["Jake", "Noah", 34, True, 40, "male"]
```



INFORMATION
TECHNOLOGY
UNIVERSITY

**What do you guys hate about C++.
Is it pointers?**

python does not have it.



INFORMATION
TECHNOLOGY
UNIVERSITY

If else statement

C++

```
If (x< 2){  
  
    cout << "x is less than 2";  
  
}  
  
else {  
  
    cout << "x is not less than 2";  
  
}
```

python

```
If x< 2:  
  
    print("x is less than 2")  
  
else:  
  
    print("x is not less than 2")
```



INFORMATION
TECHNOLOGY
UNIVERSITY

If else if statement

C++

```
If (x< 2){  
  
    cout << "x is less than 2";  
  
}  
  
else if(x> 2){  
  
    cout << "x is not less than 2";  
  
}
```

python

```
If x< 2:  
  
    print("x is less than 2")  
  
elif x > 2 :  
  
    print("x is not less than 2")
```



INFORMATION
TECHNOLOGY
UNIVERSITY

While loop

```
While (x > 2){  
    cout << x -;  
}
```

C++

```
While x > 2:  
    print(x)  
    x-=1
```

python



INFORMATION
TECHNOLOGY
UNIVERSITY

While loop

C++

```
While (x > 2){  
    cout << "x is "<< x -;  
}  
}
```

python

```
While x > 2:  
    print("x is "+ x)  
    x-=1
```

Error occur as x is int and we are concatenating it with string

Correct : print("x is "+ str(x))



INFORMATION
TECHNOLOGY
UNIVERSITY

Type Casting in python

```
x = str(3)      # x will be '3'  
y = int(3)      # y will be 3  
z = float(3)    # z will be 3.0
```

for loop

C++

```
for(int i=0; i< 10; i++){  
    cout << i;  
}
```

python

```
for i in range(start, end, Increment):  
    print(i)  
  
for i in range(0, 10):  
    print(i)
```

By default start = 0



INFORMATION
TECHNOLOGY
UNIVERSITY

for loop Increment by 2

C++

```
for(int i=0; i< 10; i+=2){  
  
    cout << i;  
  
}
```

python

```
for i in range(start, end, Increment):  
    print(i)  
  
for i in range(0, 10, 2):  
    print(i)
```

By default start = 0



INFORMATION
TECHNOLOGY
UNIVERSITY

C++

```
// Iterating over whole array
std::vector<int> v = {0, 1, 2, 3, 4, 5};
for (auto i : v)
    std::cout << i << ' ';
std::cout << '\n';
```



INFORMATION
TECHNOLOGY
UNIVERSITY

C++

```
// the initializer may be a braced-init-list
for (int n : {0, 1, 2, 3, 4, 5})
    std::cout << n << ' ';
std::cout << '\n';
```



INFORMATION
TECHNOLOGY
UNIVERSITY

C++

```
// Iterating over array
int a[] = {0, 1, 2, 3, 4, 5};
for (int n : a)
    std::cout << n << ' ';
std::cout << '\n';
```



INFORMATION
TECHNOLOGY
UNIVERSITY

C++

```
// Printing keys and values of a map
std::map <int, int> MAP({{1, 1}, {2, 2}, {3, 3}});
for (auto i : MAP)
    std::cout << '{' << i.first << ", "
                  << i.second << "}\n";
```



INFORMATION
TECHNOLOGY
UNIVERSITY

C++

```
// Printing string characters
std::string str = "Geeks";
for (char c : str)
    std::cout << c << ' ';
std::cout << '\n';
```



INFORMATION
TECHNOLOGY
UNIVERSITY

for loop with Arrays

C++

```
int cars[3] = [2, 10, 0]  
  
for(int i=0; i< 3; i++){  
  
    cout << cars[i];  
  
}
```

python

```
cars = [2, 10, 0]  
  
for i in cars:  
  
    print(i)
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Operators

&&

||

!

pow(a,b);

C++

python

and

or

Not

a**b



INFORMATION
TECHNOLOGY
UNIVERSITY

Other Operators are same as C++

Operator

Name



INFORMATION
TECHNOLOGY
UNIVERSITY

Other Operators are same as C++

Operator

Example



INFORMATION
TECHNOLOGY
UNIVERSITY

Function

C++

```
int fun(int count){  
  
    cout << "i have "<<count <<"pens";  
  
    count++;  
  
    return count;  
  
}
```

python

```
def fun(count):  
  
    print("i have "+str(count)+"pens")  
  
    count+=1  
  
    return count
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Class

C++

```
class Apple{  
    Apple(){  
    }  
    void fun(int count){  
    }  
};  
→ this
```

python

```
class Apple:  
    def __init__(self):  
        #this is constructor  
    def fun(self, count):  
        #some function
```

→ Self



INFORMATION
TECHNOLOGY
UNIVERSITY

Objects

C++

```
Apple obj;  
obj.fun(57);
```

python

```
obj = Apple()  
obj.fun(57)
```



INFORMATION
TECHNOLOGY
UNIVERSITY

Class

C++

```
class Car {  
public:  
    string brand ="abc";  
    Car(){  
    }  
};
```

python

```
class Car:  
    def __init__(self):  
        self.brand="abc"
```



INFORMATION
TECHNOLOGY
UNIVERSITY