



Introduction to Python

Live Session 3

Course : Introduction to Python - 3

Lecture On : Python Fundamentals

Instructor : Dr. Reena Duggal

Session	Topic
Session 1	Python Basics - Hello World / Data Structures / Variables / String Operations
Session 2	Python Data Structures – List / Tuple / Dictionary, Loops
Session 3	Python Programming - Conditions and Branching / Functions / Objects & Classes
Session 4	Working with Data in Python - Reading & Writing Files

In this session, we will cover..

1. Conditions/Branching
2. Functions
3. Lambda Function
4. Map, Filter & Reduce
5. Objects & Classes

Programming in Python



Variables



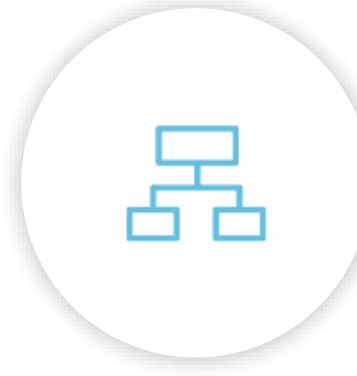
Data Types



Operators



Loops



Conditions



Functions

Conditional Statements

Python **if statement** is used to conditionally execute a block of code.

If statement contains a logical expression or condition, depending on which decisions are made.

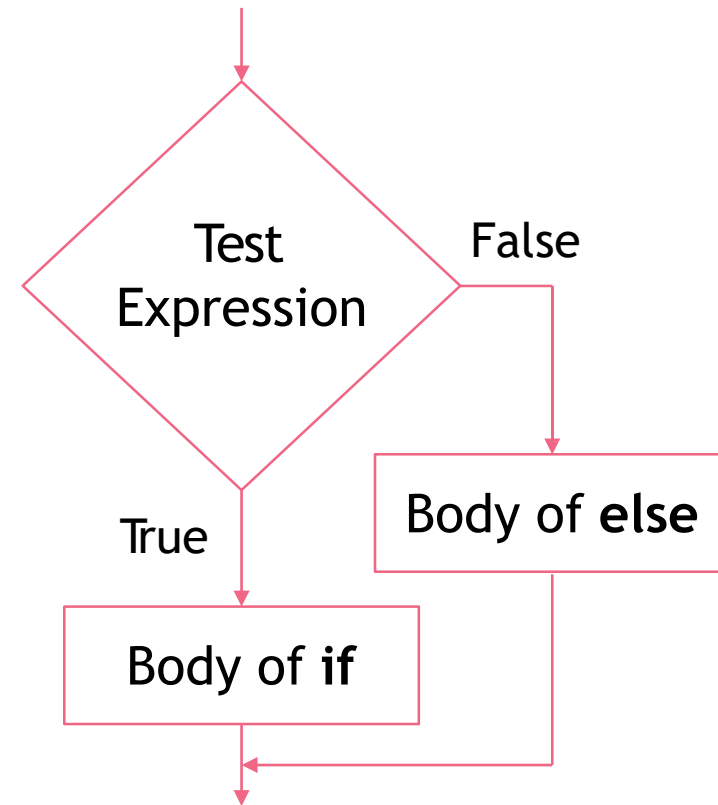


If I get at least 10 rupees, I will buy chocolate,
else I will buy candy

Conditional Statements



Syntax:
if (expression):
 statements(s)
else:
 statements(s)

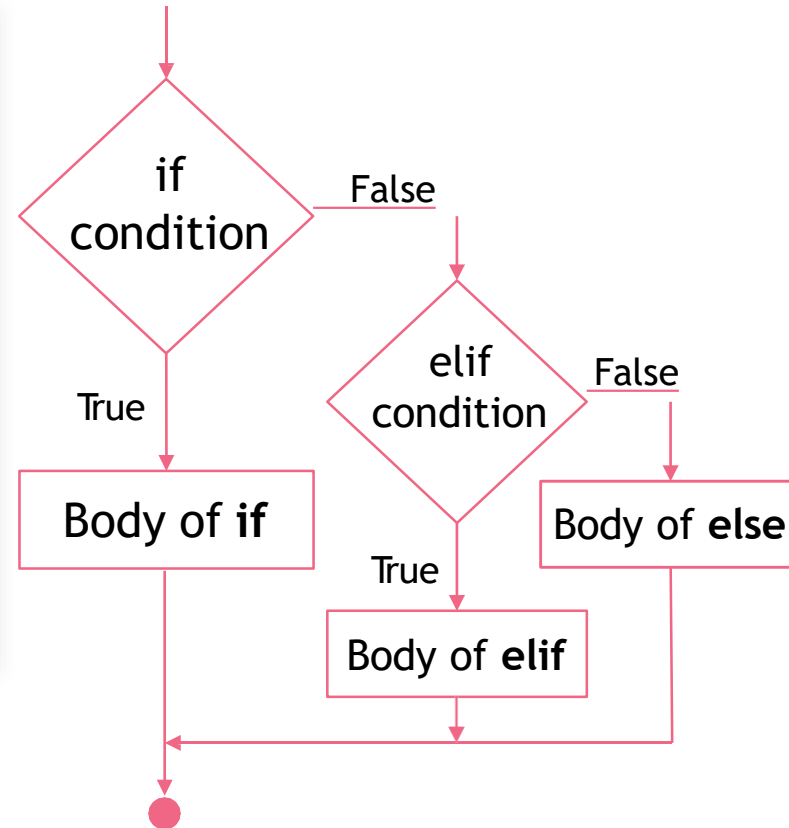


Conditional Statements

if elif
else

Syntax:

```
if (expression):  
    statement(s)  
elif (expression):  
    statement(s)  
elif (expression):  
    statement(s)  
else:  
    statement(s)
```



Conditional Statements

nested
if else

```
var = 50
if var < 200:
    print ("Expression value is less than 200")
    if var == 150:
        print ("Which is 150")
    elif var == 100:
        print ("Which is 100")
    elif var == 50:
        print ("Which is 50")
    elif var < 50:
        print ("Expression value is less than 50")
else:
    print ("Could not find true expression")

print ("Good bye!")
```

```
Expression value is less than 200
Which is 50
Good bye!
```

Poll 1

What will the following segment of code print? Try doing this verbally.

```
if (10 < 0) and (0 < -10):  
    print("A")  
elif (10 > 0) or False:  
    print("B")  
else:  
    print("C")
```

- 1) A
- 2) B
- 3) C
- 4) B and C

Poll 1(Answer)

Solution: (2)

Programming in Python



Variables



Data Types



Operators



Loops



Conditions



Functions

Lambda Function

*A lambda can have
multiple arguments
separated by commas*



lambda arguments : expression

*Every lambda begins
with the "lambda"
keyword*



*A colon precedes the
expression*

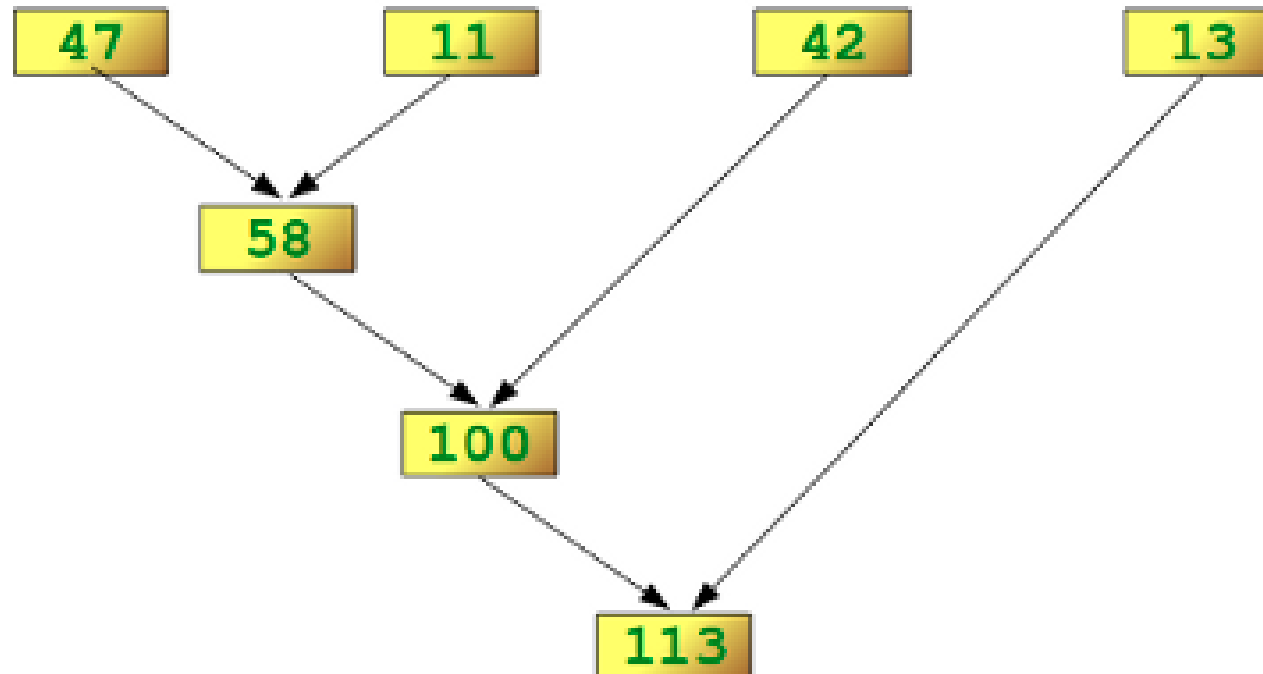


*The expression always
returns an object*

Reduce() Function

```
>>> reduce(lambda x,y: x+y, [47,11,42,13])  
113
```

The following diagram shows the intermediate steps of the calculation:



Poll 2

What will be the output of the following Python code?

```
l = [1, 2, 3, 4, 5]  
m = map(lambda x:2**x, l)  
print(list(m))
```

a) [1, 4, 9, 16, 25]

b) [2, 4, 8, 16, 32]

c) [1, 0, 1, 0, 1]

d) Error

Poll 2(Answer)

Solution: **(b)**

Objects & Classes

- A class in python is the blueprint from which specific objects are created.
- It lets you structure your software in a particular way.
- Classes allow us to logically group our data and function in a way that it is easy to reuse and a way to build upon if need to be.



(A)



(B)

Methods and Attributes in a Python Class

- Every class has a special method called **init**
- It gets called whenever a new object of that class is initialized.
- Creating a class is incomplete without some functionality.
- Functions in python are also called as Methods.

Python Class: Inheritance

- Inheritance allows us to inherit attributes and methods from the base/parent class.
- This is useful as we can create sub-classes and get all of the functionality from our parent class.
- We can overwrite and add new functionalities without affecting the parent class.

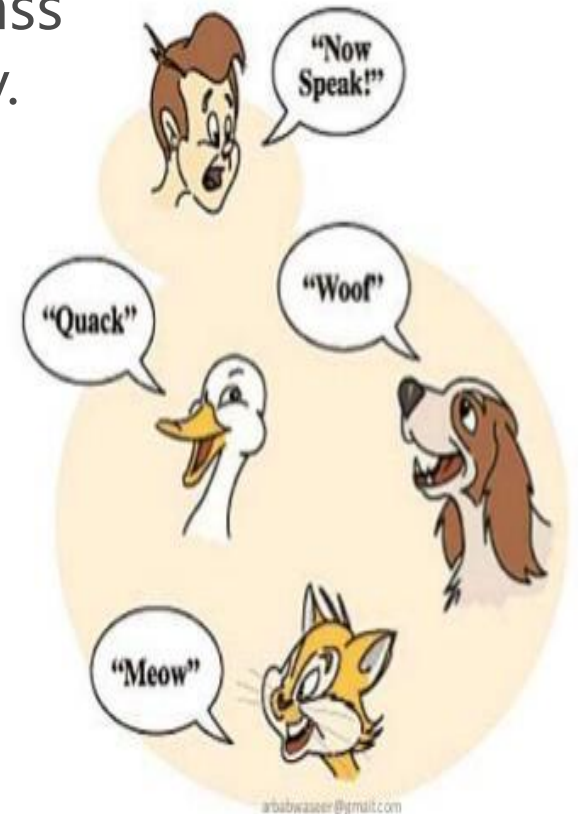
As we can see in the image, a child inherits the properties from the father. Similarly, in python, there are two classes:

1. Parent class (Super or Base class)
2. Child class (Subclass or Derived class)



Python Class: Polymorphism

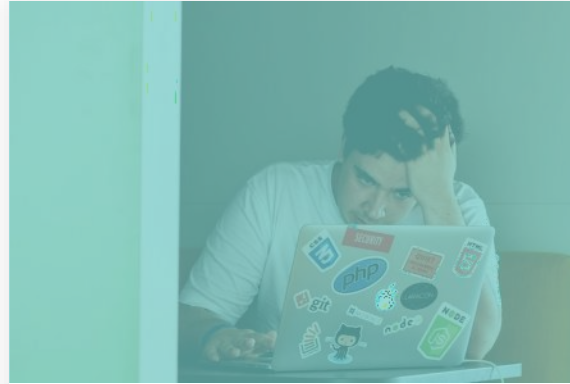
- In Greek, Polymorphism means **many**(poly) **forms**(morph).
- It means that a method or function is able to cope with different types of input.
- **Polymorphism** in **inheritance** means that if class B inherits from class A, it doesn't have to inherit everything about class A, it can do some of the things that class A does differently.



Points to Remember



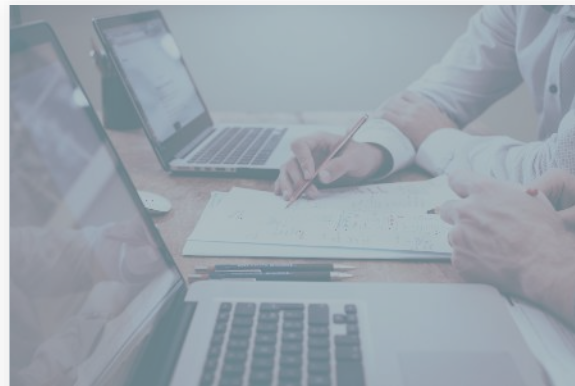
Practice Everyday



Don't overdo it



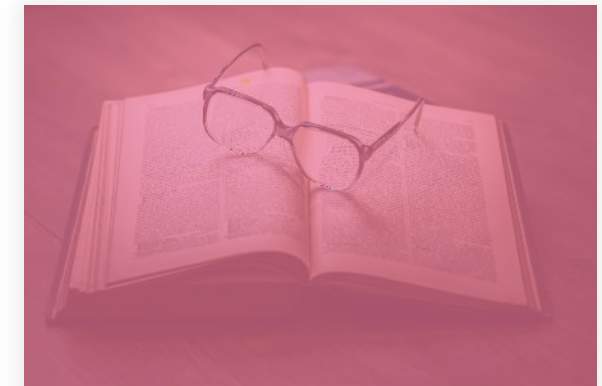
Solve it yourself



Get more information



Okay to be lost at first



Read