

INTRODUCTION TO PROGRAMMING

Introduction to Programming - Types of Languages, Memory Management

Types of languages

```
graph TD; A[Types of languages] --> B[Procedural]; A --> C[Functional]; A --> D[Object Oriented]
```

Procedural


- specifies a series of well-structured steps and procedures to compose a program.
- Contains a systematic order of statements, functions and commands to complete a task.

Functional

- Writing a program only in pure functions i.e. never modify variables, but only create new ones as an output.
- Used in situations where we have to perform lots of different operations on the same set of data, like ML.
- First class functions?

Object Oriented

- Revolves around objects
- Code + Data = Object
- Developed to make it easier to develop, debug, reuse, and maintain software.



Static vs Dynamic Languages

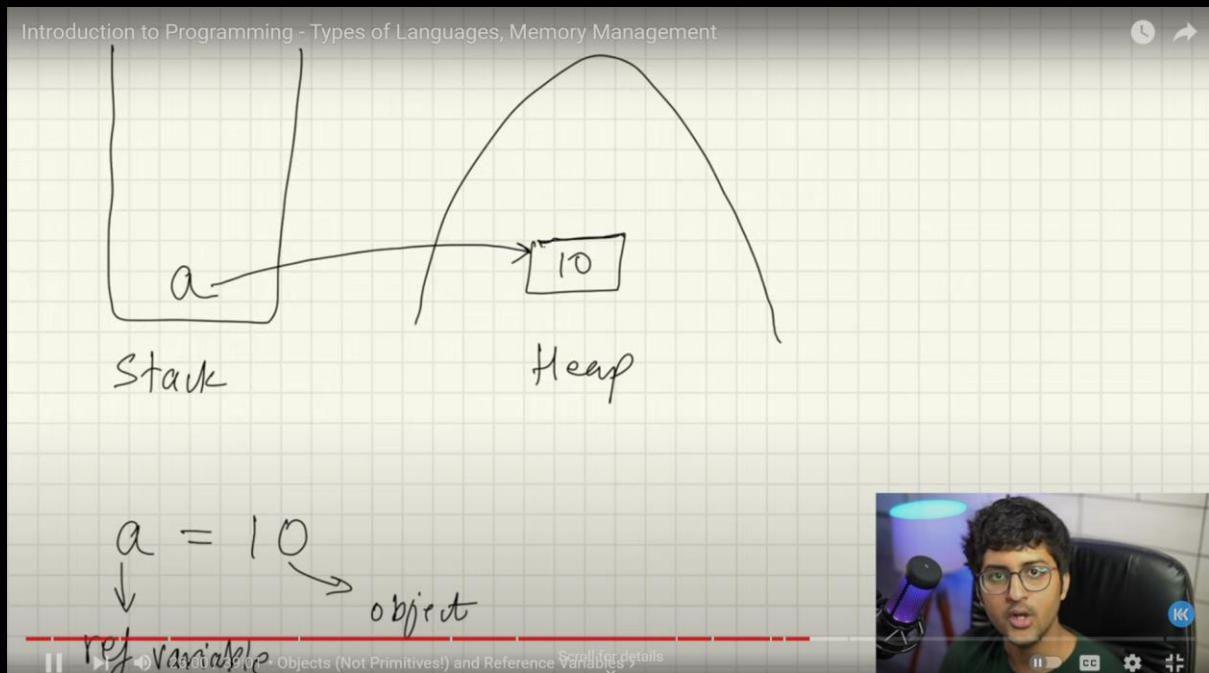
Introduction to Programming - Types of Languages, Memory Management

Static vs Dynamic Languages

Static	Dynamic
<ul style="list-style-type: none">- Perform type checking at compile time- Errors will show at compile time- Declare datatype before you use it- More control	<ul style="list-style-type: none">- Perform type checking at runtime- Error might not show till program is run- No need to declare datatype of variables- Saves time in writing code but might give error at runtime

Static Languages = C++, Java

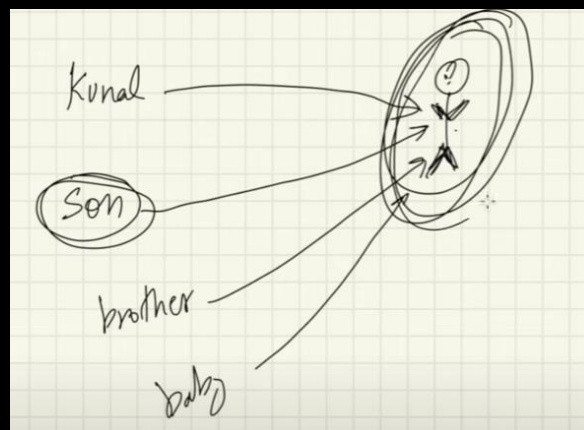
Dynamic = Python, JavaScript



Variables in stack memory and pointing towards to the heap memory

String name = "Shubham"

In these example name is reference to the object so name is stored in the stack memory and value is stored in the heap memory



More than one ref variables points to the object

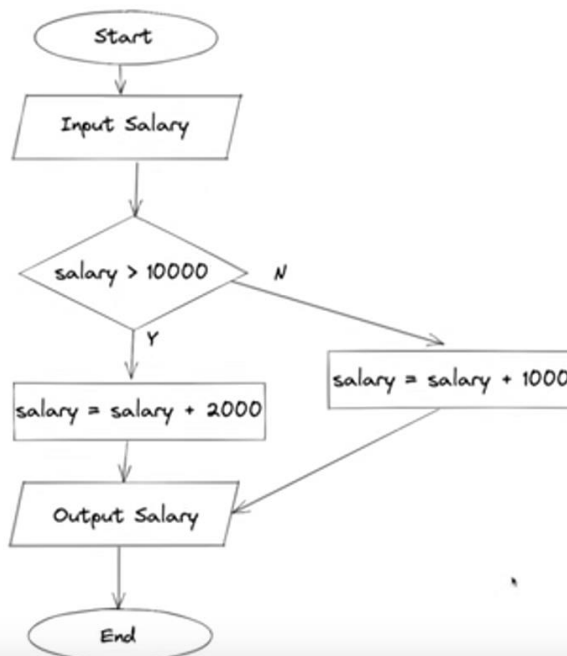
If any one of the object changes the object then it changes original objects

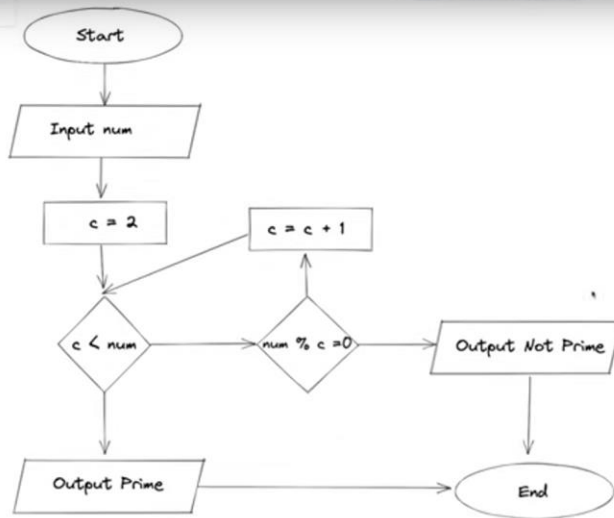
What is garbage Collection in Java?

- ⇒ Object who do not have reference variables pointing towards object
- ⇒ In that case it removes from the memory

FLOWCHARTS

Take input of a salary. If the salary is greater than 10,000 add bonus as 2000, otherwise add bonus as 1000.





```

start
input num
c = 2
while c < num:
    if num%c==0:
        output "not prime"
        exit
    c = c+1
end while
output "prime"
exit
    
```

