

COL1000: Introduction to Programming

Nuts & Bolts of Python — Conditionals

Subodh Sharma | Lec 4 | Aug 08



Reminders!

- Do **NOT** share passwords!
 - Change it frequently
- Remember to check <https://moodlenew.iitd.ac.in>
 - For lab and lecture content
 - Under Lecture Code —> SVS find the `lec3.py` and play with it!
- Reference Book: **Introduction to Computation and Programming Using Python** by John Guttag

(Python) Programming Essentials: RECAP

- **Variables:** Storage locations in memory that holds a **value**, has a **type***, and a **scope**
- **Statements:** Assignments, Conditionals/Control, Function invocations, I/O,
- **Operators:** Arithmetic, Logical, Relational, ... , Operator-Precedence
 - Operator associativity: For operators with same precedence
 - Eg: $a - b + c \rightarrow (a-b) + c$
- **Functions:** I/O functions, Built-in functions (range, min, max, input, print, etc.)

Branching Programs

- So far we have looked at straight-line programs
 - Where instructions execute one after another in the order they appear in the program
 - Branching through control statement alters the flow of execution
 - Simplest branching statement: **Conditional**

- Eg:

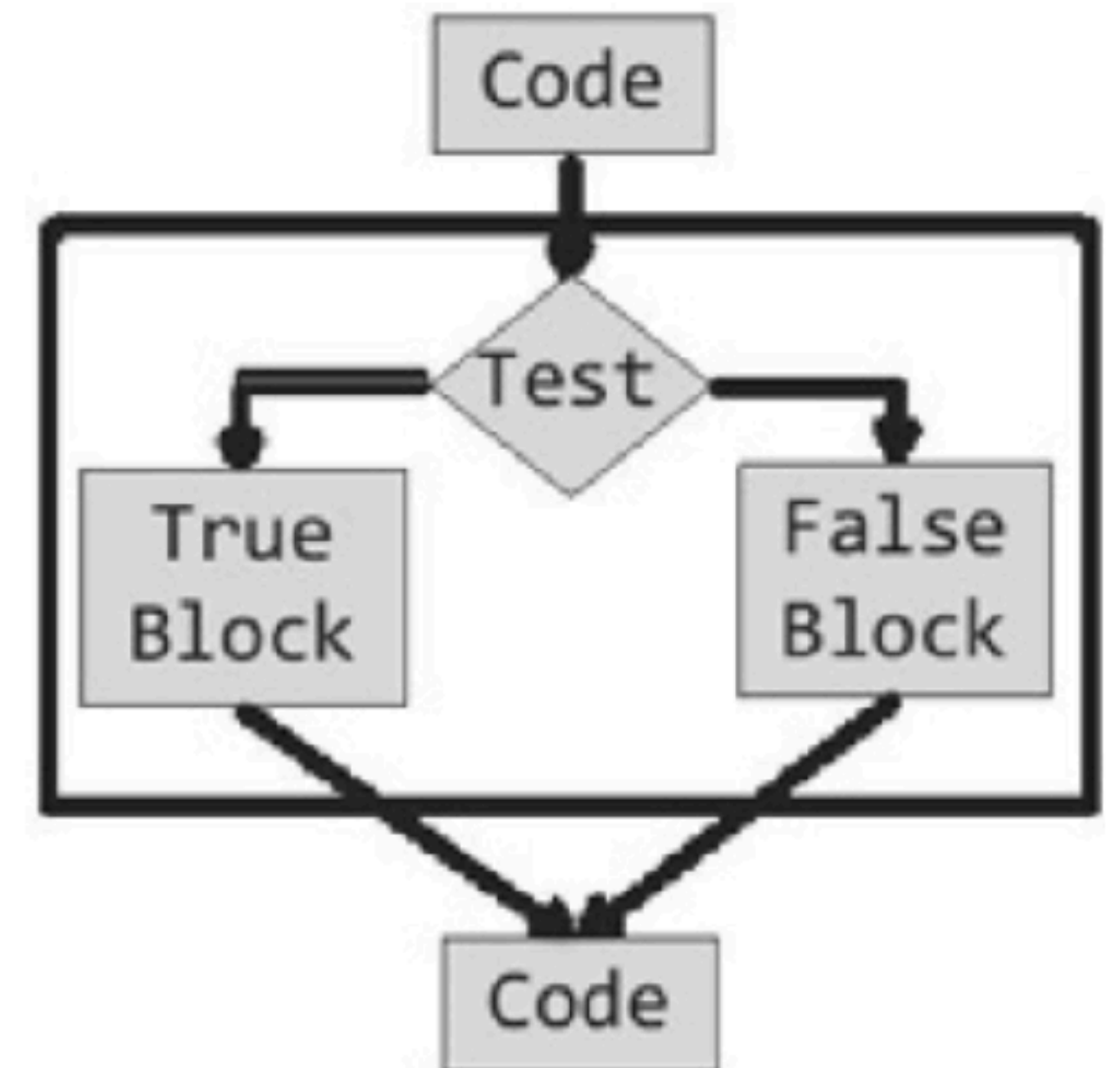
```
if x % 2 == 0:
```

Note the indentation

```
    print ("even")
```

```
else:
```

```
    print("odd")
```



Example: Find the Max, Mid, Min of 3 Ints

- Live Programming exercise!
- How do we know it is correct for **all** possible inputs
 - Use the idea of **invariants** in the form of **preconditions** and **postconditions**
- **Formal Specification:**
 - **Precondition:** An invariant that holds true at the start of the program
 - **Postcondition:** An invariant that holds true after the execution of the program

Short Circuiting in Conditionals

- The logical operators `and` and `or` can be **short-circuited**
 - That is, stop the moment final result is determined
 - This implies only partial evaluation - one could avoid unnecessary computation!
 - `A and B` # if A is falsy, then B is not evaluated
 - `A or B` # if A is truthy, then B is not evaluated
 - Also for safety — `user is not None and user.is_authenticated`
 - If `user is None` then the 2nd (unsafe) expr will not be evaluated