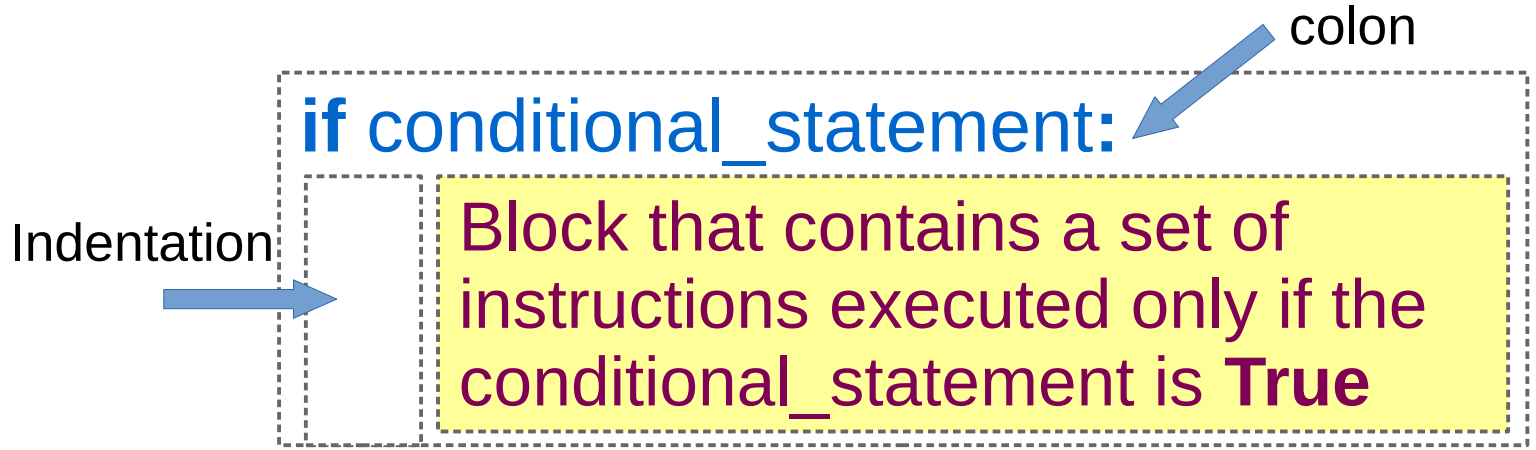
ECE-122
Chapter-2:
Elements of Programming
2.2 Conditional Statement

What is a program?

- A program is a sequence of instructions
- The details (syntax) look different depending on the programming language, but the instructions always fall into one of the following categories:
  - **input**: Get data from the keyboard, a file, the network, or some other device.
  - **output:** Display data on the screen, save it in a file, send it over the network, etc.
  - **computation**: might be something mathematical, evaluate an expression and assign a new variable, etc.
  - **conditional statement**: Check for certain conditions and select the actions.
  - **repetition:** Perform some action repeatedly, usually with some variation.
- **That's pretty much all there is to it.**
- Programming is the process of breaking a large, complex tasks into smaller subtasks until they can be performed with one of these basic instructions. There are many ways to do this.

Conditional Statements

- So far, we've viewed programs as sequences of instructions that are followed one after the other.
- However, we need to be able to alter the sequential flow of a program to suit a particular situation.
- One of the most important notion in programming is the possibility to ask questions that can be answered only by Yes (True) or No (False)
- This can easily be accomplished using the if statement



Conditional Statements: if

- Conditional_statement returns a boolean **True** or **False**
- It can also be the result of a simple comparison between two values or expressions. The relational operators are:

Python	Mathematics	Meaning
<	<	Less than
<=	≤	Less than or equal to
==	=	Equal to
>=	≥	Greater than or equal to
>	>	Greater than
!=	≠	Not equal to

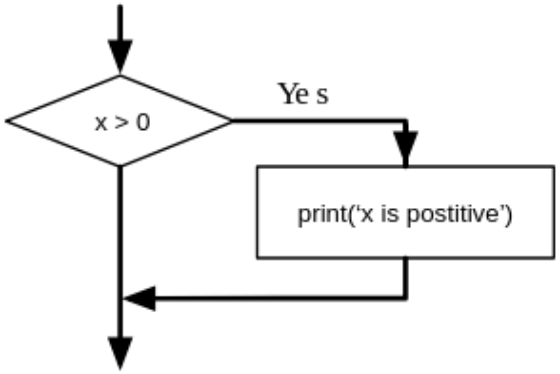
Conditional Statements: if

- Examples:

```
is_student=True
if is_student: #statement below runs only if True
    print("You are a student")
```

→ You are a student

```
x=5
if x>0:
    print("x is positive")
```



→ x is positive

```
x=int(input("Choose x: "))
if x>0:
    print("x is positive")
    x=x+1
print("x is",x)
```

→ Choose x: 3
x is positive
x is 4

→ Choose x: -3
x is -3

Conditional Statements: if

- Multiple conditional statements can be combined using the boolean operators **and, or**

```
x=int(input("Choose x: "))
if x>0 and x%2==0:
    print("x is positive and even")
```

Choose x: 0

Choose x: 4
x is positive and even

```
x=int(input("Choose x: "))
if (x>0 and x%2==0) or x==0:
    print("x is positive and even, or zero")
```

Choose x: 0
x is positive and even, or zero

Remark:

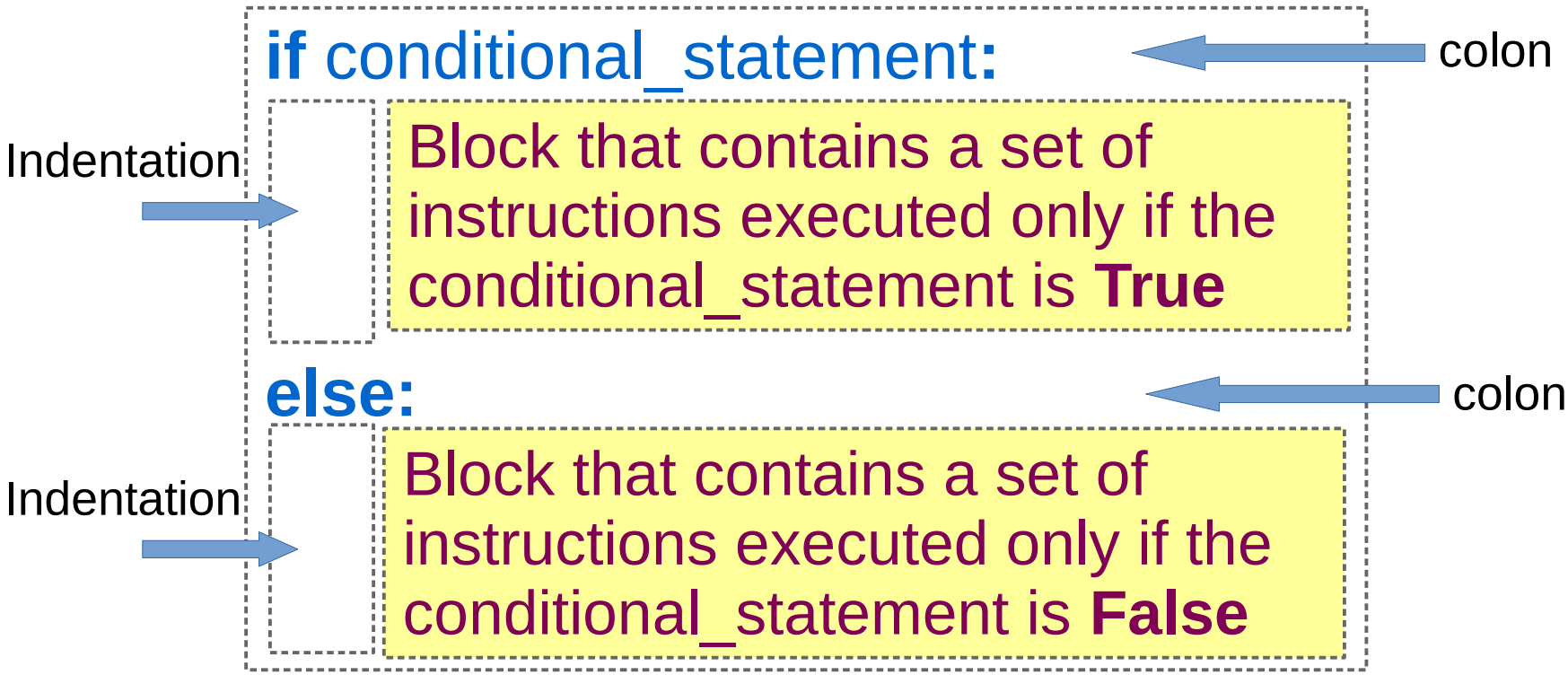
- Evaluation from left to right.
- python uses short-circuit strategy- stop evaluating conditions as early as possible

```
x,y=0,2
if (y/x>1 and x!=0):
    print("enter inside")
```

```
x,y=0,2
if (x!=0 and y/x>1):
    print("enter inside")
```

Conditional Statements: if-else

- Two ways decision is also possible using the **else** instruction



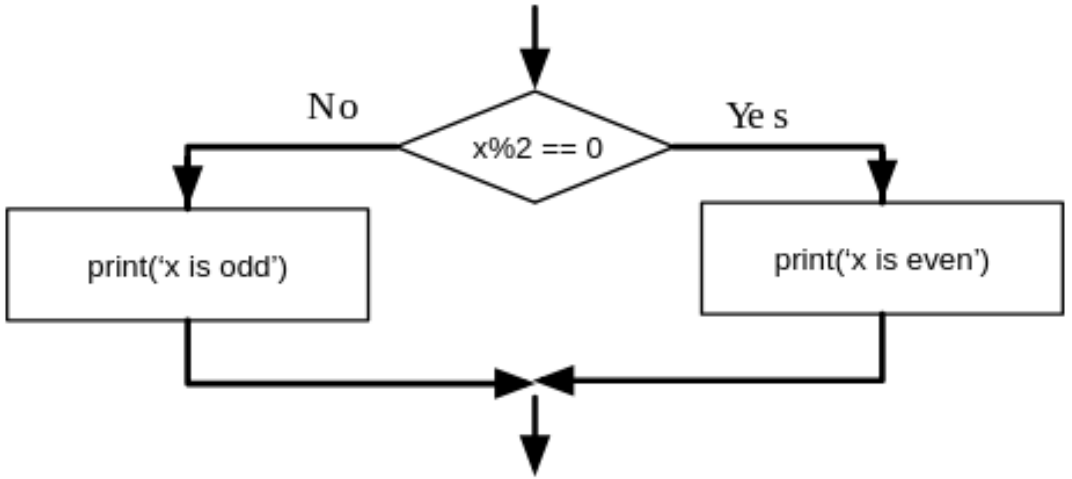
Conditional Statements: if-else

- Examples:

```
is_student=False
if is_student: #statement below runs only if True
    print("You are a student")
else:          #statement below runs only if False
    print("You are not a student")
```

You are not a student

```
x=int(input("Choose x: "))
if x%2==0:
    print("x is even")
else:
    print("x is odd")
```



Conditional Statements: if-else

- Conditional statements can be nested- Example (depth of 2):

```
x=int(input("Choose x: "))
if x>0:
    if x%2==0:
        print("x is positive and even")
    else:
        print("x is positive and odd")
else:
    print("x is negative or zero")
```

Choose x: 4
x is positive and even

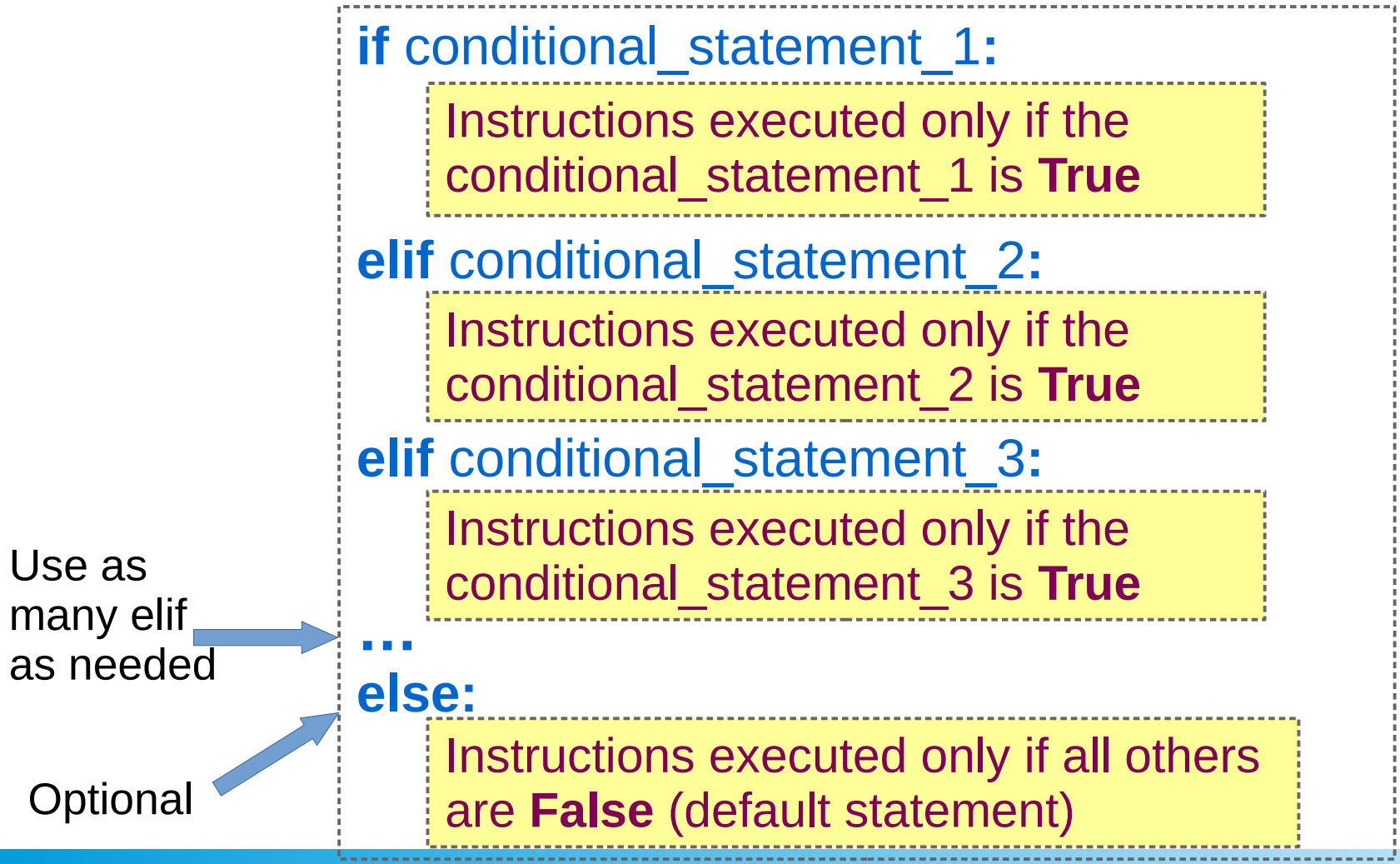
Choose x: 1
x is positive and odd

Choose x: -1
x is negative or zero

- Nested conditional statements are perfectly valid but can become very difficult to read if the depth is greater than 2 or 3
- If we needed to make a five-way decision using nesting, the if-else statements would be nested four levels deep!
- Good idea to avoid many nested conditionals if possible

Conditional Statements: elif

- Chained conditionals in Python achieves multi-decisions, by combining an **else** followed immediately by an **if** into a **elif**.



Conditional Statements: elif

- Examples:

```
is_student=True
like_cake= False

if is_student and like_cake:
    print("You are a student and you like cake!")
elif is_student and not(like_cake):
    print("You are a student and you do not like cake!")
elif not(is_student) and like_cake:
    print("You are not a student and you like cake!")
else:
    print("You are not a student and you do not like cake!")
```

Ternary if

- Example: finding the max between 2 numbers

```
# Python code
if num1>num2:
   maxval=num1
else:
   maxval=num2
```

vs

```
/* C code here */
if (num1>num2)
   maxval=num1;
else
   maxval=num2;
```

- For **simple two decision ways**, C code uses the ternary operator

```
/* C code here */
maxval = (num1 > num2) ? num1 : num2;
```

- In Python, it can also be done in a more intuitive way

```
/* Python code here */
maxval = num1 if (num1 > num2) else num2
```

- It is actually, more than an abbreviation, it can be used as expression

```
maxval = (num1 if (num1 > num2) else num2)* 4 - 2.5
```