Basic Program Structure
More on Selection (Branching) - PYTHON

Lesson Objectives

At the end of this lesson, you should be able to:

- Apply the following selection (branching) structures in Python:
  - IF-ELSE statement
  - IF-ELIF-ELSE statement
  - Nested IF statement
- Explain all the basic arithmetic operators in Python
- Use the basic arithmetic operators to solve problems

Topic Outline

IF-ELSE Statement

Arithmetic Operators in Python

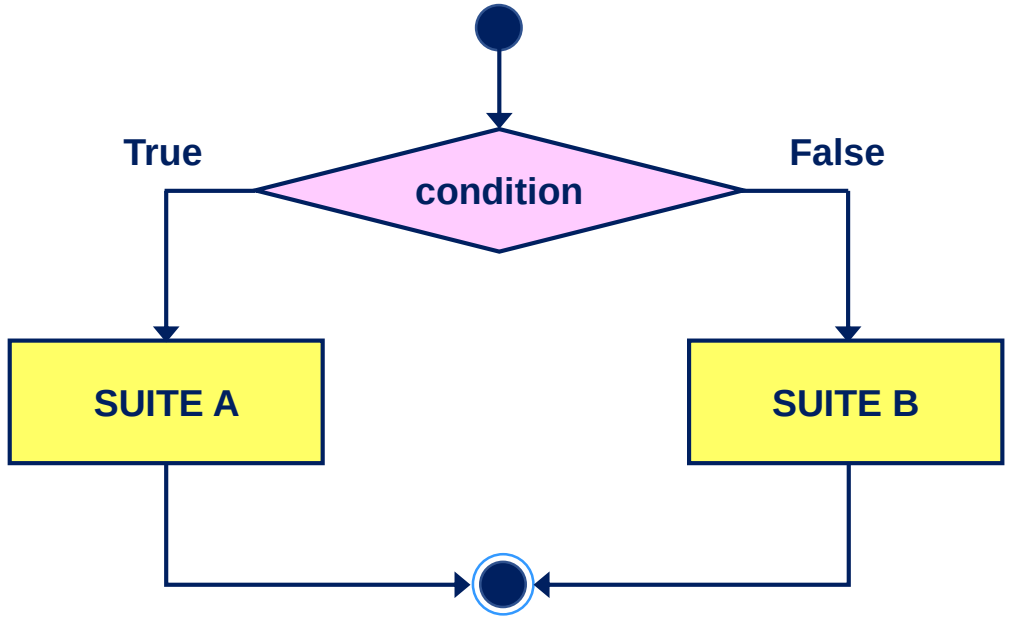
IF-ELIF-ELSE Statement

Nested IF Statement

IF-ELSE Statement (recall)

Pseudocode

```
IF condition is True
THEN
      SUITE A
ELSE
      SUITE B
END IF
...
```



IF-ELSE Statement - General Python Syntax

Syntax

```
if condition:
    indentedStatementBlockForTrueCondition
else:
    indentedStatementBlockForFalseCondition
```

These statement blocks can have any number and type of statements.

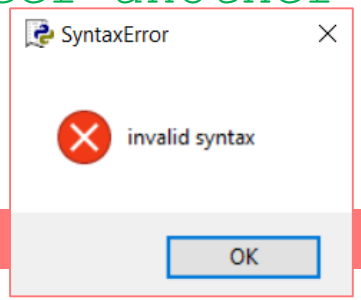
IF-ELSE Statement

Examples

```
if a > b:
        print("a > b")
else:
        print("a <= b")
```

In Python, a colon marks the start of a block or suite.

```
a = (int)(input("please enter an integer:"))
b = (int)(input("please enter another integer:"))
if a > b:
        print ("a > b")
        print ("a > b")
else
        print ("a <= b")
```



Message

Do not forget the colon. Otherwise, it causes a syntax error!

IF-ELSE Statement (Cont'd)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

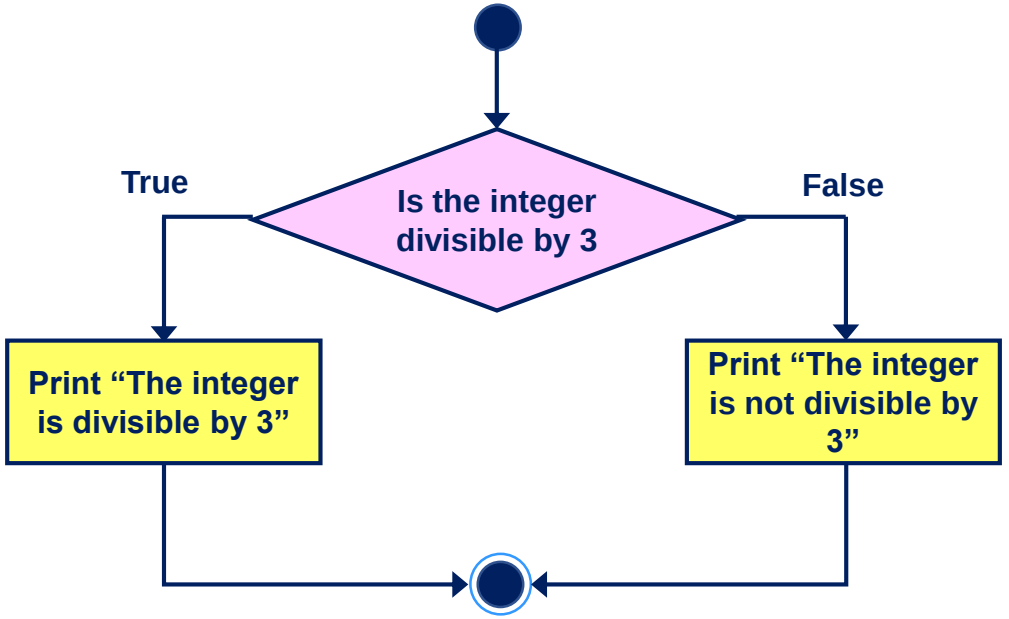
Example

```
import math
radiusString = input("Enter the radius of your circle:")
radiusFloat = float(radiusString)
if radiusFloat > 0:
    circumference = 2 * math.pi * radiusFloat
    area = math.pi * radiusFloat * radiusFloat

    print("The circumference of your circle is: ",
    circumference,\
              ", and the area is:", area)
else:
  print("radius must be positive")
  print("please try again")
```

Application of IF-ELSE Statement

Determine if an Integer is Divisible by 3



Quick Try: Intelligent Guess

How to determine whether m is divisible by n?

How to determine whether the remainder of m/n is zero in Python?

Modulus

Operator	Meaning	Syntax	Example (x = 8, y = 3)
%	Modulus operator – returns the remainder of the division of left operand by the right	x % y (remainder of x/y)	8 % 3 = 2

```
m = 8
n = 3
if m % n == 0:
        print(n, "is a divisor of", m)
else:
        print(n, "is not a divisor of", m)
```

Quick Check

Today is Tuesday. After 53 days, what day will it be?

Quick Check: Answer

Today is Tuesday. After 53 days, what day will it be?

Answer **Saturday**

```
>>> 2 + 53 % 7
6
```

```
    7
  ---
7 ) 53
    49
    --
     4
```

Arithmetic Operators in Python

Operator	Meaning	Operator	Meaning
+	• Add two operands • Unary plus	%	Modulus - remainder of the division of left operand by the right
–	• Subtract the right operand from the left • Unary minus	//	Floor division (integer division) - the resultant value is a whole integer, although the result's type is not necessarily int
*	Multiply two operands	**	Exponent - the left operand is raised to the power of the right operand
/	Floating point division: divide the left operand by the right one (**always results in a `float`**)		

Arithmetic Operators in Python (Cont'd)

Operator	Meaning	Example	Output (x = 8, y = 3)
+	• Add two operands • Unary plus	• x + y • +x	• 11 • 8
–	• Subtract the right operand from the left • Unary minus	• x – y • –x	• 5 • –8
*	Multiply two operands	x * y	24

Arithmetic Operators in Python (Cont'd)

Operator	Meaning	Example	Output (x = 8, y = 3)
/	Floating point division: divide the left operand by the right one (**always results in a `float`**)	x / y	2.666666666666665
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)	2
//	Floor division (integer division) - the resultant value is a whole integer, although the result's type is not necessarily int	x // y	2
**	Exponent - the left operand is raised to the power of the right operand	x ** y (x to the power of y)	512

Note: Division in Python

Differences between Python Versions | In Python 3.x: **Integer/integer** → **Float**

- 8/4 → 2.0
- 8/3 → 2.6666666666666665

```
>>> 8.0/4.0
2.0
>>> 8/4
2.0
>>> int(8/4)
2
>>> int(8.0/4.0)
2
```

A trick in Python 3:

```
>>> 8//4
2
>>> 8/3
2.6666666666666665
>>> 8//3
2
```

Floor division

Note: Division by Zero

- 1 / 0 ➡ Runtime error
- How about 1 % 0? Try it yourself.

```
>>> 1 / 0
Traceback (most recent call last):
      File "<pyshell#12>", line 1, in <module>
      1 / 0
ZeroDivisionError: division by zero
>>> 1 % 0
Traceback (most recent call last):
      File "<pyshell#13>", line 1, in <module>
      1 % 0
ZeroDivisionError: integer division or modulo by zero
```

Quick Check

Grace has $25. She wants to buy movie tickets at $3 each. Which of the following statements calculates the maximum number of tickets that she can purchase?

A. 25 / 3 B. 25 % 3 C. 25 // 3 D. 25 %% 3

Quick Check: Answer

Grace has $25. She wants to buy movie tickets at $3 each. Which of the following statements calculates the maximum number of tickets that she can purchase?

A. 25 / 3

B. 25 % 3

remainder

C. 25 // 3

quotient

D. 25 ** 3

Answer **C. 25 // 3**

Mixed Operations

- We have seen **8 / 3** and **8.0 / 4.0**.
- How about **8 / 3.0**? Here, there are **different data types**, i.e., **int and float**.
  - This is called a mixed operation.
  - Actually, there is no mixed operation.
  - Data are **implicitly** converted.

Python will **automatically convert** the data to the **most detailed** result.

- Thus, 8 → 8.0 and the result is 2.6666666.
- Detail: int < float.

Note: It is more detailed but (maybe) less precise.

Order of Calculation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

General mathematical rules apply:

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction

Note: Always use parentheses if in doubt.

2 * 2 ** 2 vs. (2 * 2) ** 2

Augmented Assignments

- These operations are **shortcut**.
- They make the code easier to read.

Shortcut	Equivalence
`myInt += 2`	`myInt = myInt + 2`
`myInt -= 2`	`myInt = myInt - 2`
`myInt /= 2`	`myInt = myInt / 2`
`myInt *= 2`	`myInt = myInt * 2`
`myInt %= 2`	`myInt = myInt % 2`

Quick Check

What is the output of the following code?

```
x = 1.5
y = 2
x *= y
print (x)
```

A. 1.5 B. 2 C. 3.0 D. 3

Quick Check: Answer

What is the output of the following code?

```
x = 1.5
y = 2
x *= y
print (x)
```

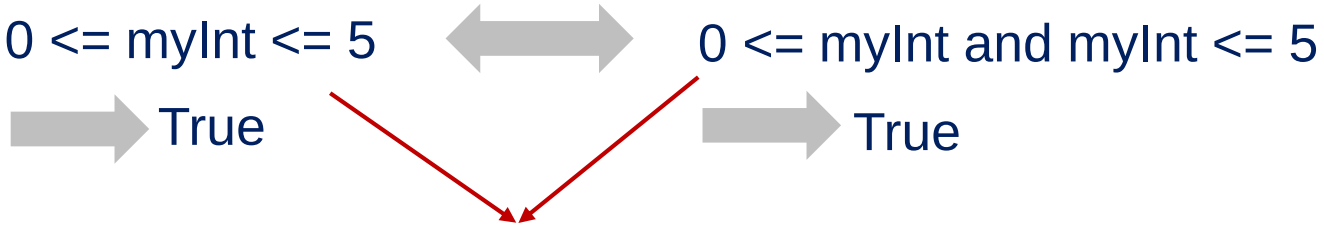
A. 1.5 B. 2 C. 3.0 D. 3

```
x ← 1.5
y ← 2
x *= y  ⟺  x = x * y
x = 1.5 * 2.0
x = 3.0
```

Answer **C. 3.0**

Chained Comparisons

- **In Python** (but not in most languages), chained comparisons work just like you would expect in a mathematical expression.
- Say, **myInt** has a value of **5**:



Both have the same meaning, i.e., implicit "**and**".

```
grade = 87
if(80 < grade <= 90):
        print('very good')
if(90 < grade <= 100):
        print('excellent')
```

More Example

0 <= myInt <= 5 > 10

False ← Just apply each operator to compare its two **neighbouring** values and then "and" the results.

Precedence of Relational and Arithmetic Operators

Increase in priority

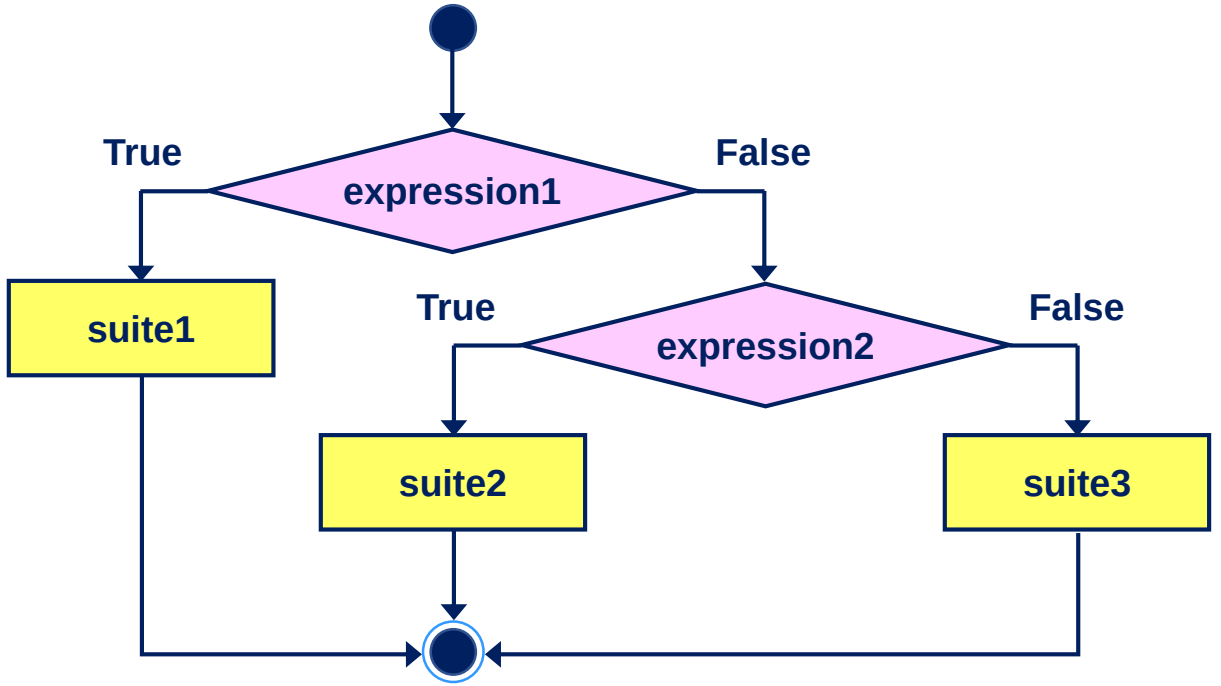
Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

IF-ELIF-ELSE Statement - General Python Syntax

Syntax

```
if expression1:
    suite1
elif expression2:
    suite2
else:
    suite3
```

Inside the False block, you may have another IF statement(s).



elif == else if

IF-ELIF-ELSE Statement (Cont'd)

Example

```
if a > b:
        print("a > b")
        print("case 1 here")
elif a < b:
        print("a < b")
        print("case 2 here")
else:
        print("a == b")
        print("case 3 here")
```

In Python, indentation defines a block.
If you continue to use the same amount of indentation, it is still in the same block.

IF-ELIF-ELSE Statement (Cont'd)

Pseudocode

```
IF condition1 is True THEN
        DO SUITE A
ELIF condition2 is True THEN
        DO SUITE B
ELIF condition3 is True THEN
        DO SUITE C
ELSE
        DO SUITE D # conditions 1,2,3 are all false
END IF
……
```

Whole statement

You may have many, many ELIF blocks.

IF-ELIF-ELSE Statement (Cont'd)

Further Example

```
mark = float(input('what is your mark?'))

if 90 <= mark <= 100:
        print('Congratulations. You received an A.')
elif 80 <= mark < 90:
        print('You received a B.')
elif 70 <= mark < 80:
        print('You received a C.')
elif 60 <= mark < 70:
        print('You received a D.')
else:
        print('oops, not good. Wish you can be better next time.')
```

Output:
What is your mark? 95
Congratulations. You received an A.

Output:
What is your mark? 55
oops, not good. Wish you can be better next time.

Quick Check

Is the following Python code correct?

```
mark = float(input('what is your mark?'))

if 90 <= mark <= 100:
        print('Congratulations. You received an A.')
if 80 <= mark < 90:
        print('You received a B.')
if 70 <= mark < 80:
        print('You received a C.')
if 60 <= mark < 70:
        print('You received a D.')
else:
        print('oops, not good. Wish you can be better next time.')
```

Quick Check: Answer

Is the following Python code correct?

```
mark = float(input('what is your mark?'))

if 90 <= mark <= 100:
        print('Congratulations. You received an A.')
if 80 <= mark < 90:
        print('You received a B.')
if 70 <= mark < 80:
        print('You received a C.')
if 60 <= mark < 70:
        print('You received a D.')
else:
        print('oops, not good. Wish you can be better next time.')
```

Output:
What is your mark? 95
Congratulations. You received an A.
oops, not good. Wish you can be better next time.

Answer No

Program Clinic

```
if 90 <= mark <= 100:
        print('Congratulations. You
received an A.')
elif 80 <= mark < 90:
        print('You received a B.')
elif 70 <= mark < 80:
        print('You received a C.')
elif 60 <= mark < 70:
        print('You received a D.')
else:
        print('oops, not good. Wish you can
be better next time.')
```

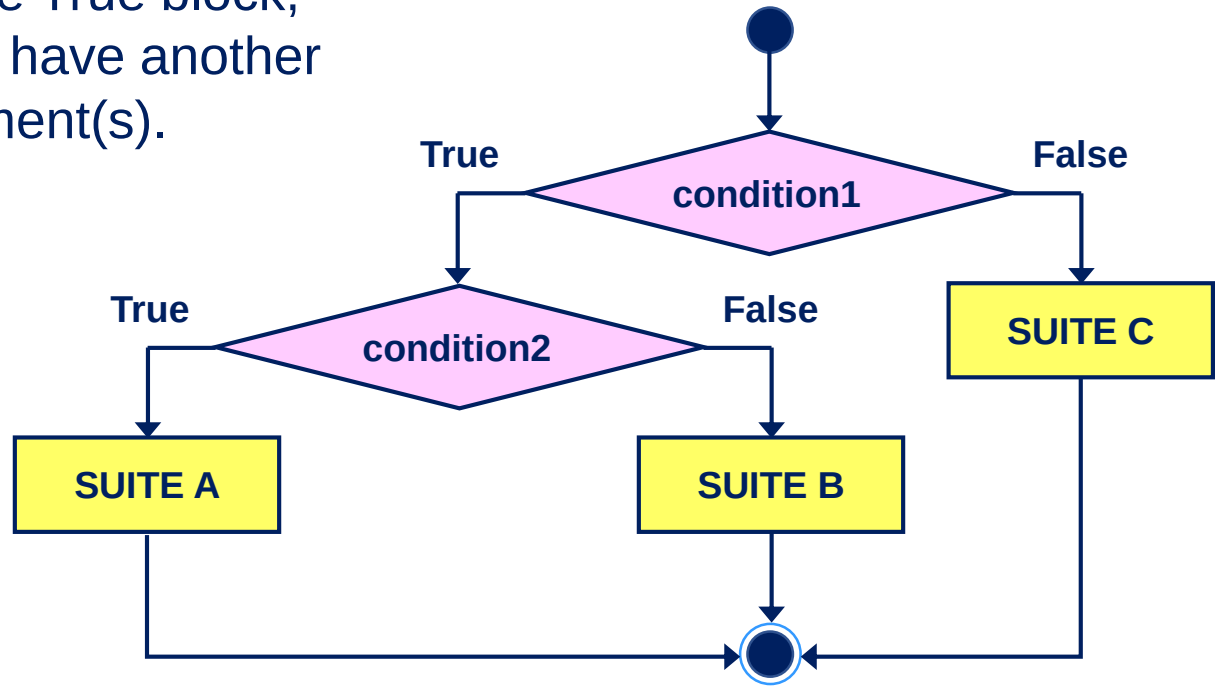
```
if 90 <= mark <= 100:
        print('Congratulations. You
received an A.')
if 80 <= mark < 90:
        print('You received a B.')
if 70 <= mark < 80:
        print('You received a C.')
if 60 <= mark < 70:
        print('You received a D.')
else:
        print('oops, not good. Wish you can
be better next time.')
```

Nested IF Statement

Syntax

```
if condition1:
    if condition2:
        SUITE A
    else:
        SUITE B
else:
    SUITE C
```

Inside the True block, you may have another IF statement(s).



Nested IF Statement (Python)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Example

```
if a >= b:
        if a >= c:
                print("maximum value is ", a)
        else:
                print("maximum value is ", c)
else:
        if b >= c:
                print("maximum value is ", b)
        else:
                print("maximum value is ", c)
```

Proper indentation is needed for the second level.

Note: There is no new syntax here.

Nested IF Statement (Cont'd)

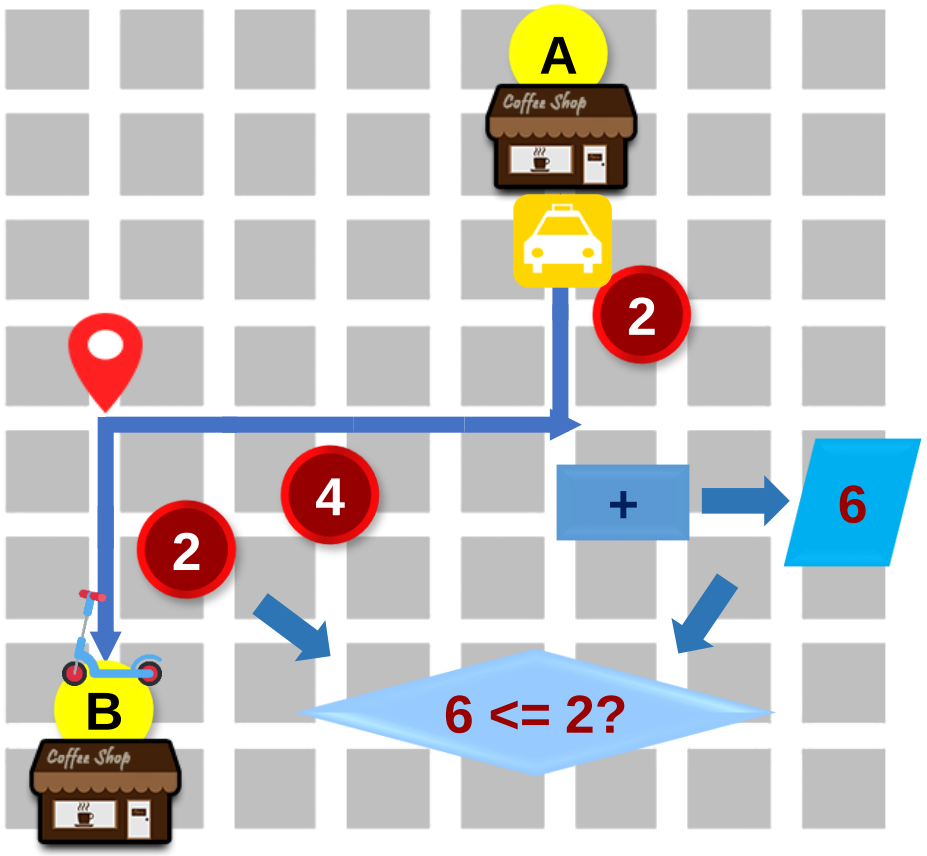
Further Example

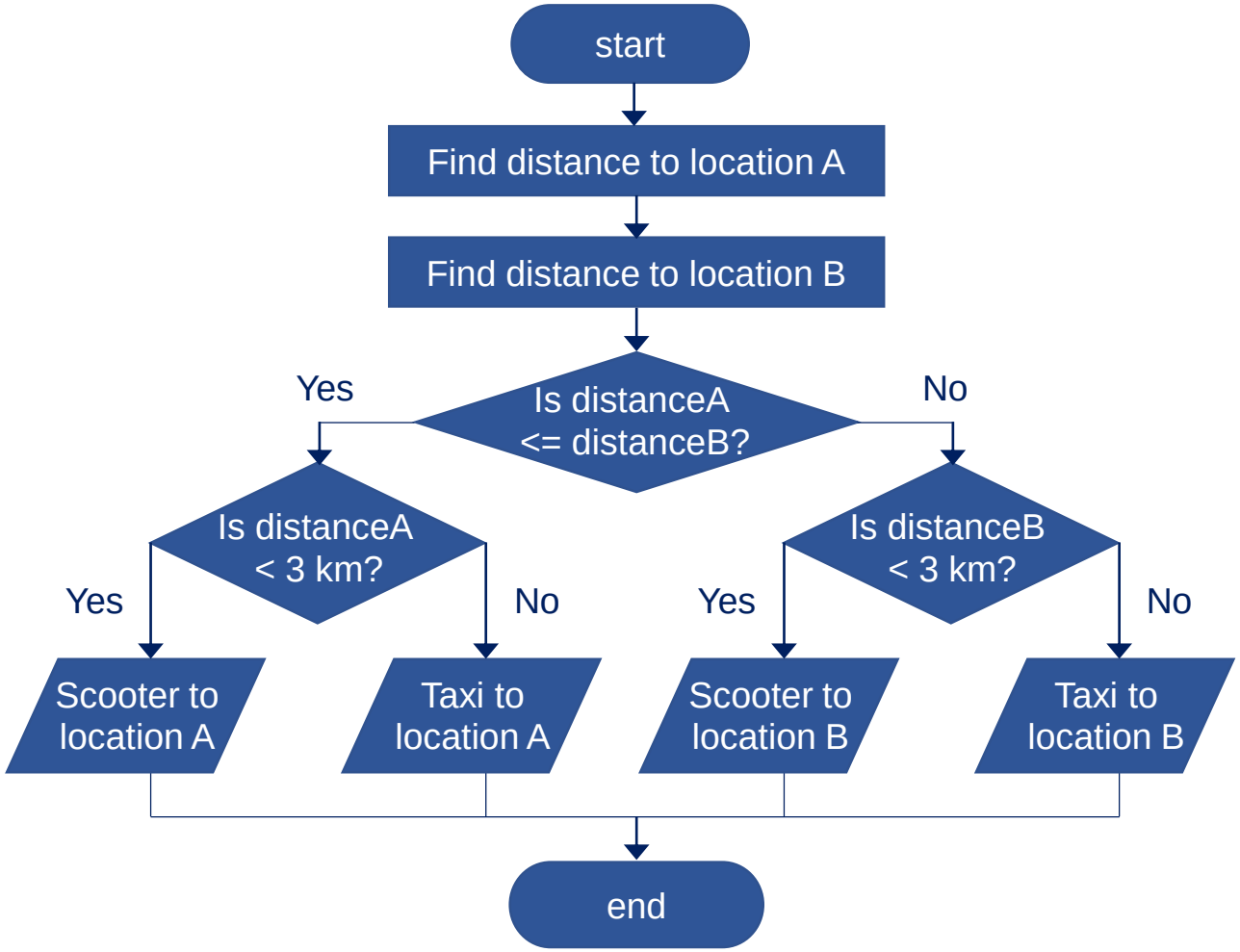
```
mark = float(input('what is your mark?'))

if mark >= 60:
        if mark >= 70:
                if mark >= 80:
                        if mark >= 90:
                                print('Congratulations. You received an A.')
                        else:
                                print('You received a B.')
                else:
                        print('You received a C.')
        else:
                print('You received a D.')
else:
        print('oops, not good. Wish you can be better next time.')
```

Scenario 6: Decide Transportation to the Nearer Coffee Shop

Recall





Scenario 6: Python Code

```
#My first Python application
#Date: 11/04/2018
#if-else; if-elif-else
horizonDistA = (int)(input("Read horizonDistA in meters"))
vertDistA = (int)(input("Read vertDistA in meters"))
distA = horizonDistA + vertDistA
print("dist from home to A is ", distA, "m")
horizonDistB = (int)(input("Read horizonDistB in meters"))
vertDistB = (int)(input("Read vertDistB in meters"))
distB = horizonDistB + vertDistB
print("dist from home to B is ", distB, "m")

if distA <= distB:
        if distA < 3000:
                print("You are suggested to ride a e-scooter to location A")
        else:
                print("You are suggested to take a taxi to location A")
elif distB < 3000:
                print("You are suggested to ride a e-scooter to location B")
else:
                print("You are suggested to take a taxi to location B")
print("Thank you for using this application")
```

Summary

More on Selection (Branching) SYNTAX

IF-ELSE

```
if condition:
    indentedStatementBlockForTrueCondition
else:
    indentedStatementBlockForFalseCondition
```

IF-ELIF-ELSE

```
if expression1:
      suite1
elif expression2:
      suite2
else:
      suite3
```

(No NEW syntax)

Nested IF

```
if condition1:
    if condition2:
       SUITE A
    else:
       SUITE B
else:
    SUITE C
```

Summary (Cont'd)

Relational and Arithmetic Operations

Increase in priority

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

References for Images

No.	Slide No.	Image	Reference
1	All Examples		Survey icon [Online Image]. Retrieved April 18, 2018 from https://pixabay.com/en/survey-icon-survey-icon-2316468/.
2	All Python codes		Python Logo [Online Image]. Retrieved April 18, 2018 from https://pixabay.com/en/language-logo-python-2024210/.
3	21, 36		By User:Bobarino - Made by following Information.png, CC BY-SA 3.0, retrieved April 18, 2018 from https://en.wikipedia.org/w/index.php?curid=9180601.
4	6, 39		Search [Online Image]. Retrieved April 18, 2018 from https://pixabay.com/en/database-search-database-search-icon-2797375/.