Beginner Programming Fundamentals With Python

Module 3

Decision Structures and Boolean Logic

Ali Samanipour

Samani Academy

Apadana OTC

Jan 2022

Module Roadmap

1 The If Statement

2 The If-else Statement

3 Nested Decision Structures

4 The If-elif-else Statement

5 Logical Operators

Ali Samanipour
Samani.Academy@gmail.com

Control vs Sequence structure

A **control structure** is a logical design that controls the order in which a set of statements execute.

A **sequence structure** is a set of statements that execute in the order in which they appear.

The if Statement

The **if statement** is used to **create a decision structure**, which allows a program to have more than one path of execution. <u>The if statement causes one or more statements to execute only when a Boolean expression is true</u>

We are determining **whether the condition** <Cold outside> is **<u>true</u>** or **<u>false</u>**.

If this condition is true, the action Wear a coat is performed. If the condition is false, the action is skipped.
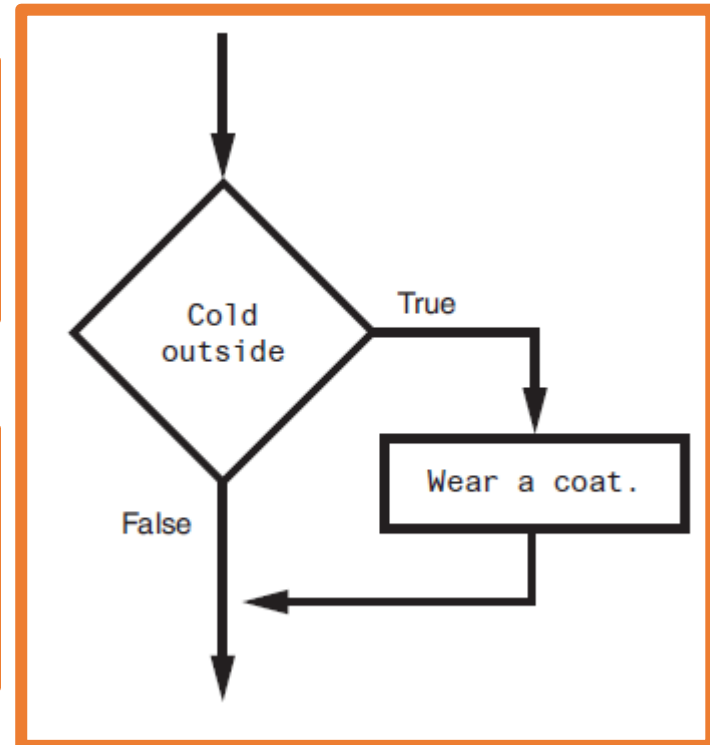
Boolean Expressions and Relational Operators

The **expressions that are tested by the if statement** are called *Boolean expressions*.

Typically, the Boolean expression that is tested by an if statement is formed with a relational operator. A ***relational operator*** determines whether a specific relationship exists between two values.

Relational Operators

Use to compare two values, the result is True or False

Used in conditional statements (if , while)

Conditional check using Boolean operators

==	!=	>=
>	<	<=

Ali Samanipour
Samani.Academy@gmail.com

Relational Operators

Operator	Name	Example
==	Equality	a == b
!=	Not equal	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b

```python
1  x = 1
2  y = 4
3  #We want to check the boolean expersion evaluation
4  print(x == y)
5  print(x != y)
6  print(x <= y)
7  print(x >= y)
8  print(x <  y)
9  print(x >  y)
```

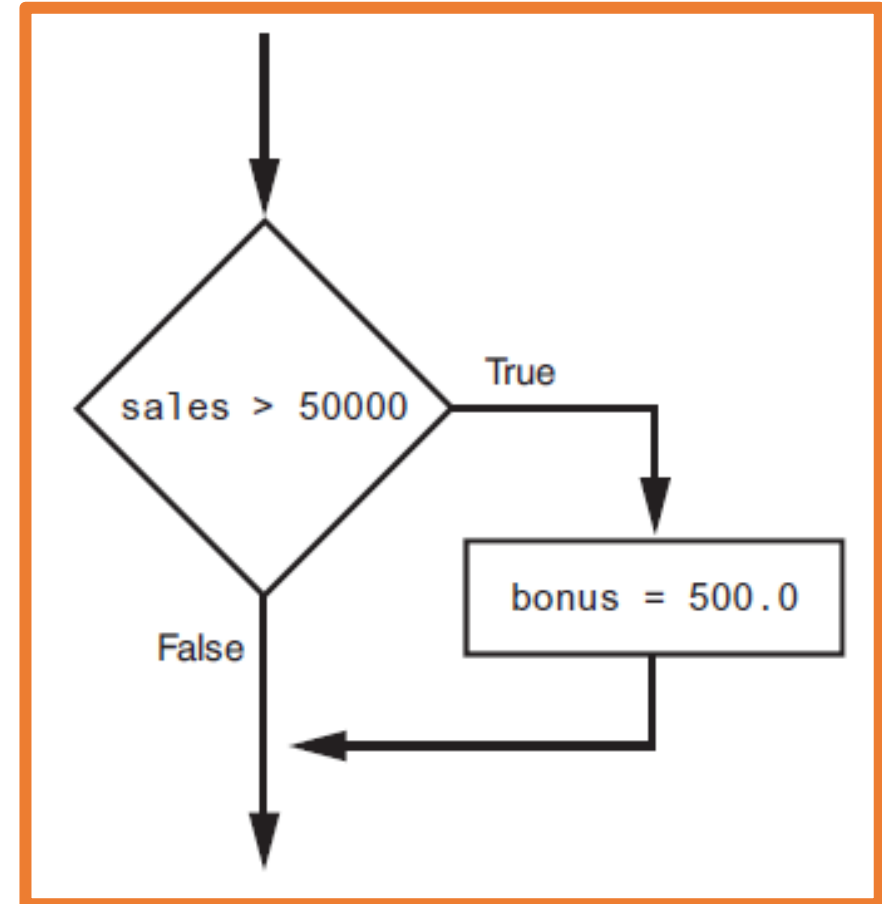Lets try!

If and Block Statement in Python

In Python, we use the if statement to write a **single alternative decision** structure.

```
if condition:
        statement
        statement
        etc.
```

block is simply a set of statements that belong together as a group. Notice in the general format that all of the statements in the block are indented.
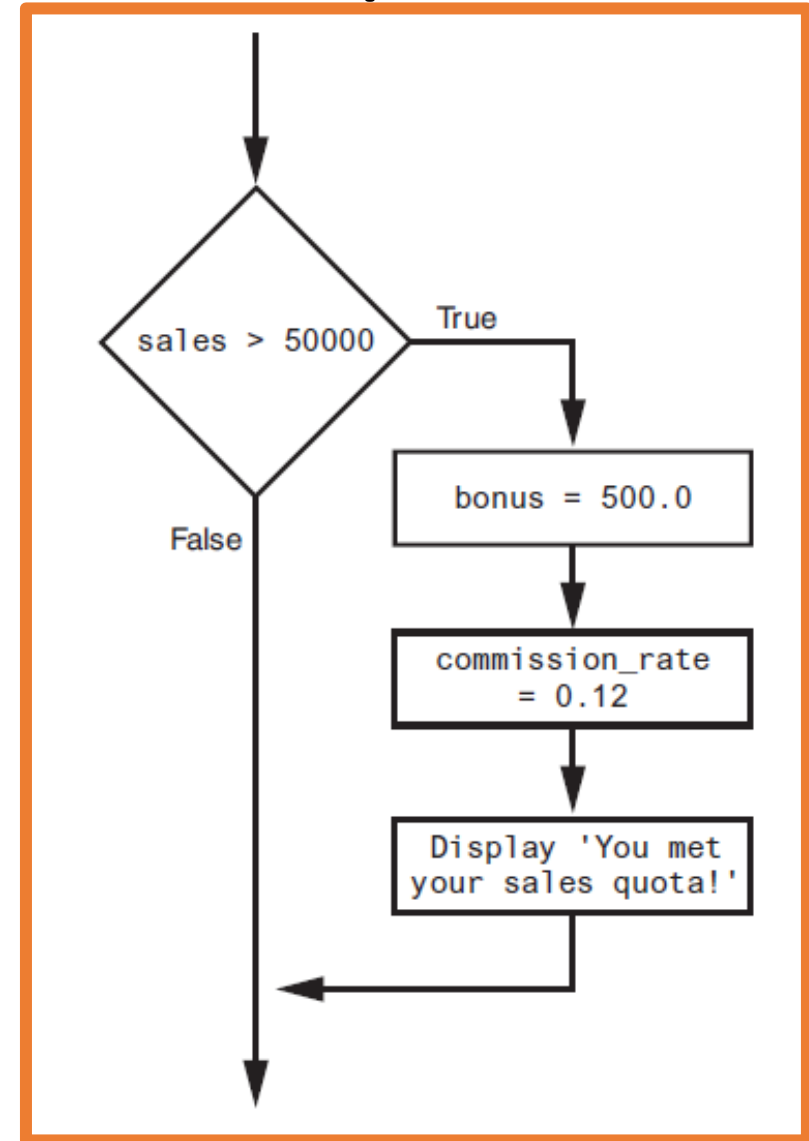
If Statement in Python Example

```
if sales > 50000:
    bonus = 500.0
```

If Statement in Python Example

```python
if sales > 50000:
    bonus = 500.0
    commission_rate = 0.12
    print('You met your sales quota!')
```

Module Roadmap

1 The If Statement

2 The If-else Statement

3 Nested Decision Structures

4 The If-elif-else Statement

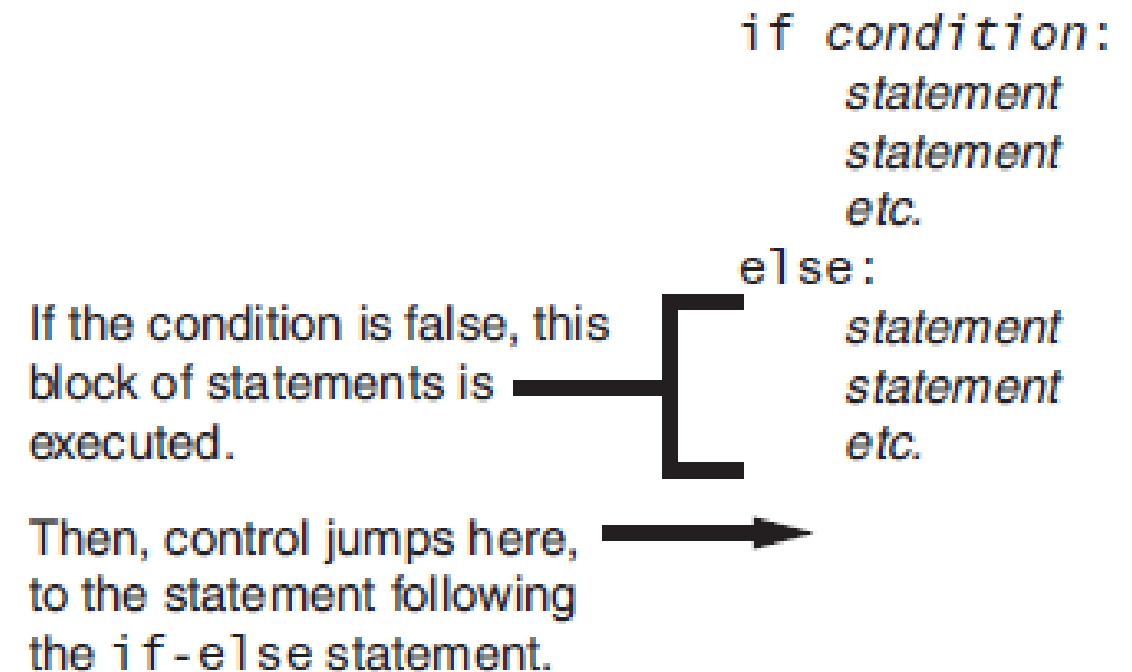**5** Logical Operators

Ali Samanipour
Samani.Academy@gmail.com

The if-else Statement

An **if-else** statement will execute one block of statements if its condition is true, or another block if its condition is false.

dual alternative decision structure, which has two possible paths of execution one path is taken if a condition is true, and the other path is taken if the condition is false

```
if condition:
    statement
    statement
    etc.
else:
    statement
    statement
    etc.
```

The if-else Statement Execution Flow

```
                                    if condition:
If the condition is true, this          statement
block of statements is                  statement
executed.                               etc.
                                    else:
                                        statement
Then, control jumps here,               statement
to the statement following              etc.
the if-else statement.
```

```
                                    if condition:
                                        statement
                                        statement
                                        etc.
                                    else:
If the condition is false, this         statement
block of statements is                  statement
executed.                               etc.

Then, control jumps here,
to the statement following
the if-else statement.
```
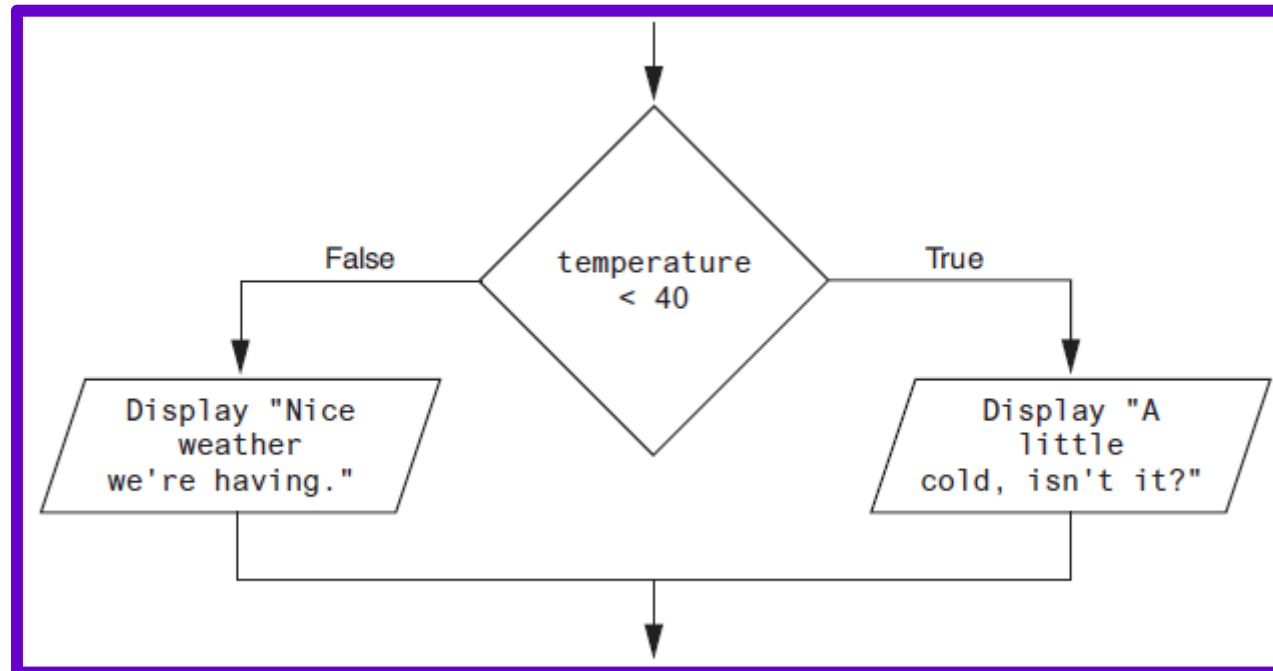
Ali Samanipour
Samani.Academy@gmail.com

The if-else Statement Example

Align the `if` and `else` clauses.

```
if temperature < 40:
    print("A little cold, isn't it?")
    print("Turn up the heat!")
else:
    print("Nice weather we're having.")
    print("Pass the sunscreen.")
```

The statements in each block must be indented consistently.

Lets try!

```python
1  password = "123V34a@"
2  def check_pass():
3      input_pass = input("Please enter password :")
4      if input_pass == password:
5          print("Password is correct ")
6          print("You are logged in successfully")
7      else:
8          print("Entered password in incorrect")
9          print("Please try agin")
10
11 check_pass()
```

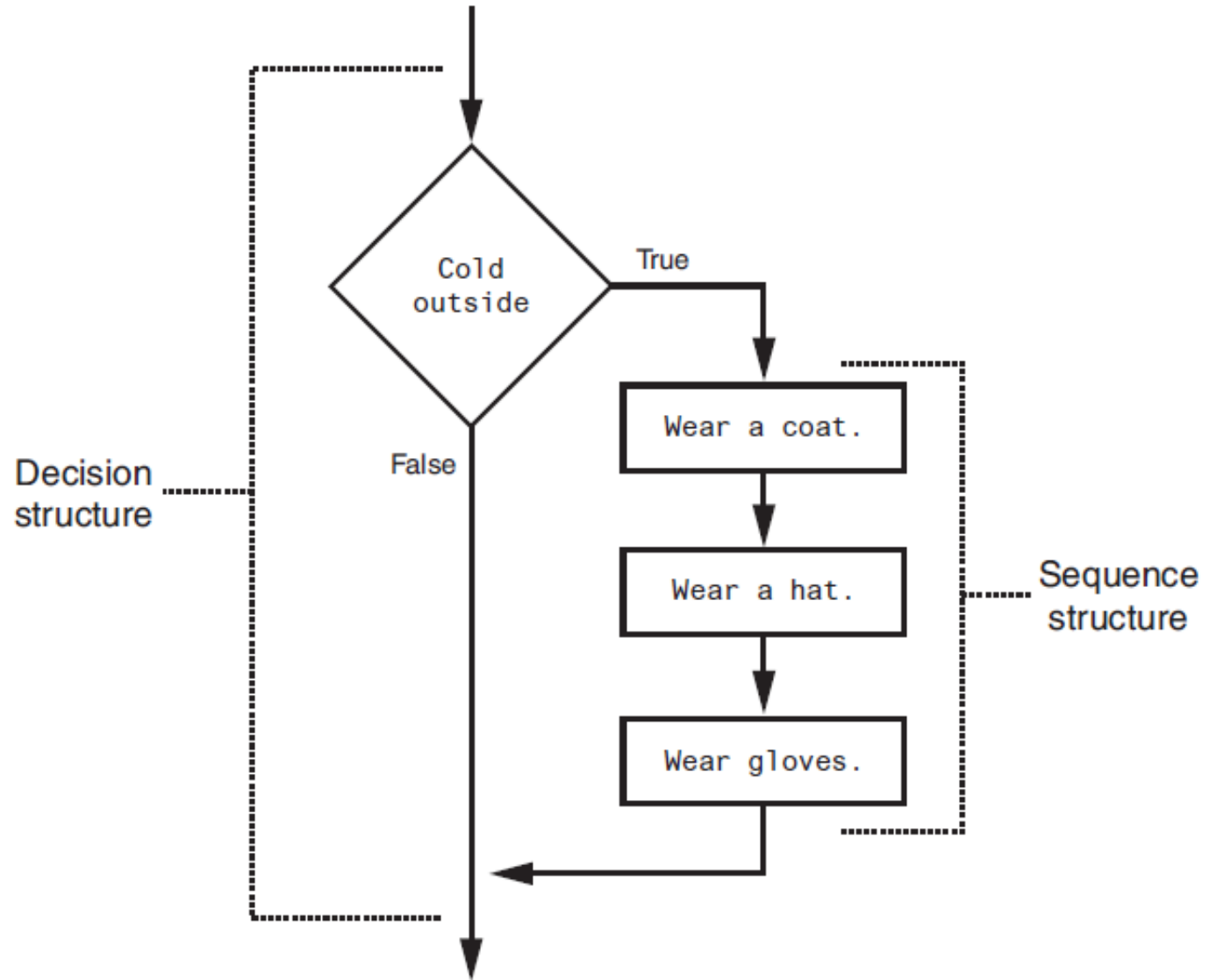⚠️ This is just an example, you should **not** use this code in real systems to check passwords

Ali Samanipour
Samani.Academy@gmail.com

Module Roadmap

1   The If Statement

2   The If-else Statement

3   Nested Decision Structures

4   The If-elif-else Statement

5   Logical Operators

Nested Decision Structures

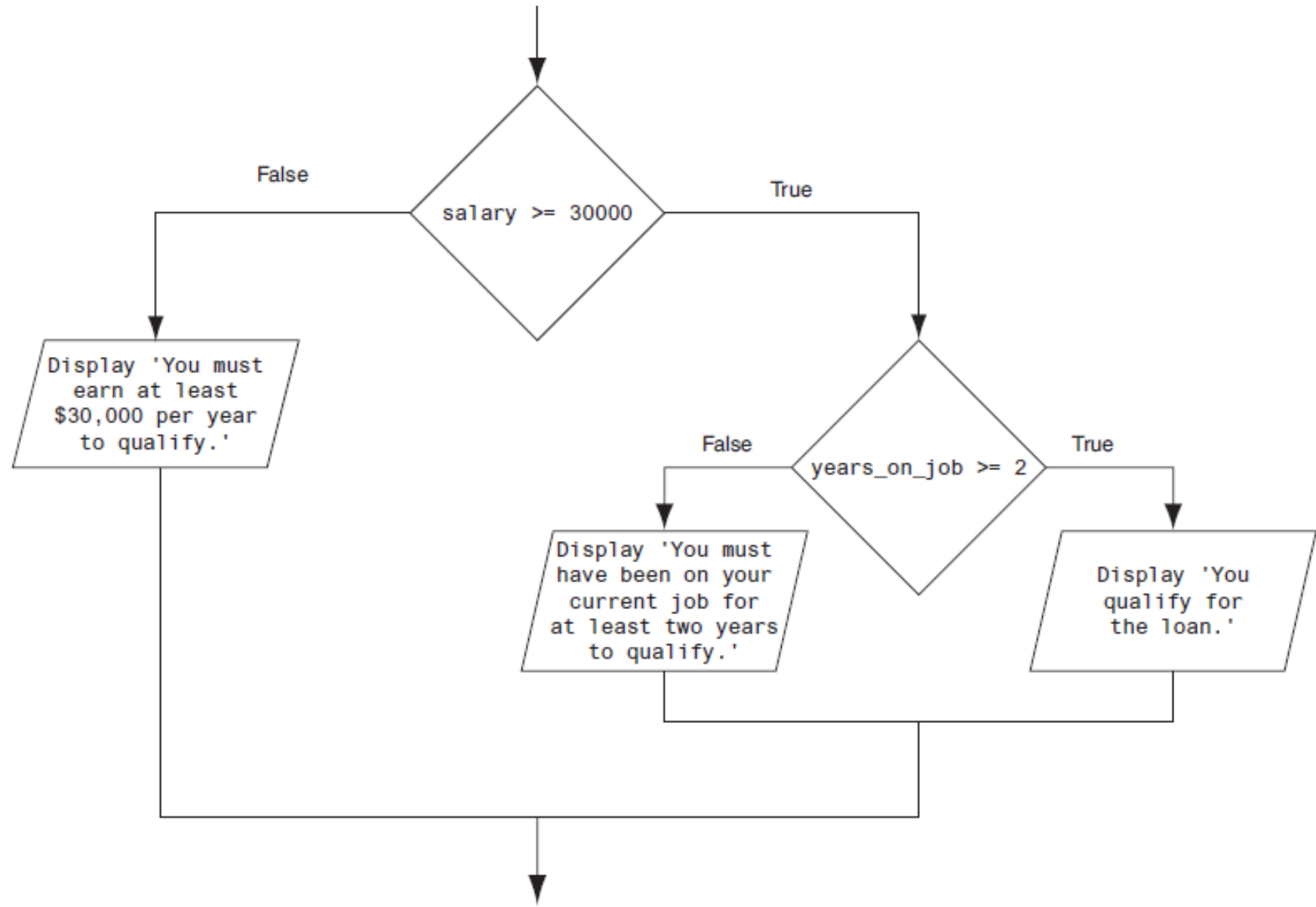To test more than one condition, a decision structure can be nested inside another decision structure
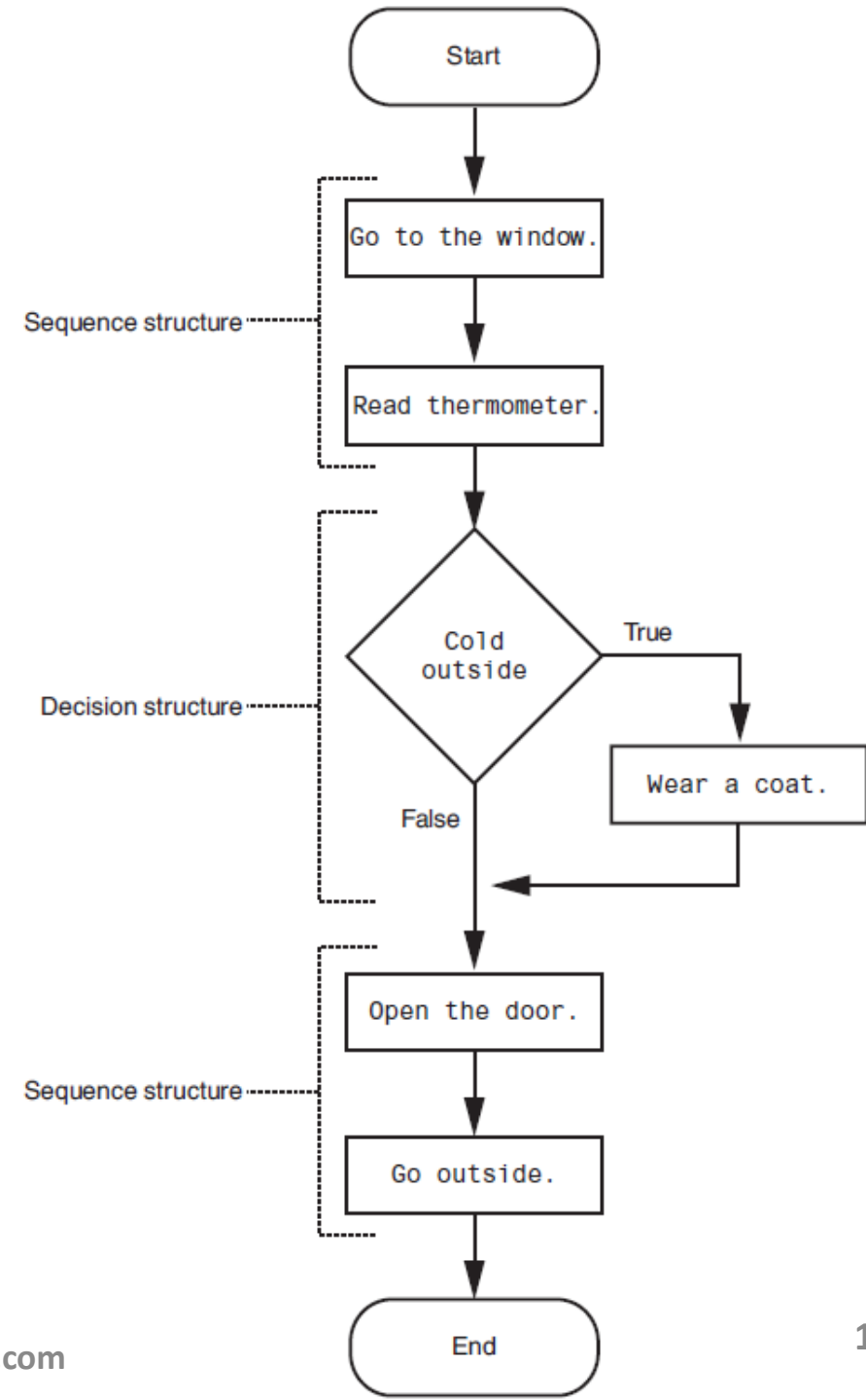
A sequence structure nested inside a decision structure

A nested decision structure

salary >= 30000

False → Display 'You must earn at least $30,000 per year to qualify.'

True → years_on_job >= 2

False → Display 'You must have been on your current job for at least two years to qualify.'

True → Display 'You qualify for the loan.'

A nested decision structure (Code Example)

```python
1  # This program determines whether a bank customer
2  # qualifies for a loan.
3  MIN_SALARY = 30000.0 # The minimum annual salary
4  MIN_YEARS = 2 # The minimum years on the job
5  # Get the customer's annual salary.
6  salary = float(input('Enter your annual salary: '))
7  # Get the number of years on the current job.
8  years_on_job = int(input('Enter the number of ' + 'years employed: '))
9  # Determine whether the customer qualifies.
10 if salary >= MIN_SALARY:
11     if years_on_job >= MIN_YEARS:
12         print('You qualify for the loan.')
13     else:
14         print(f'You must have been employed 'f'for at least {MIN_YEARS} 'f'years to qualify.')
15 else:
16     print(f'You must earn at least $'f'{MIN_SALARY:,.2f} 'f'per year to qualify.')
```

Ali Samanipour
Samani.Academy@gmail.com

21

Module Roadmap

1 The If Statement

2 The If-else Statement

3 Nested Decision Structures

4 The If-elif-else Statement

5 Logical Operators

Ali Samanipour
Samani.Academy@gmail.com

The if-elif-else Statement

```
if condition_1:
    statement
    statement
    etc.
elif condition_2:
    statement
    statement
    etc.

Insert as many elif clauses as necessary . . .

    else:
        statement
        statement
        etc.
```

```
if score >= A_SCORE:
    print('Your grade is A.')
elif score >= B_SCORE:
    print('Your grade is B.')
elif score >= C_SCORE:
    print('Your grade is C.')
elif score >= D_SCORE:
    print('Your grade is D.')
else:
    print('Your grade is F.')
```

Switch-case Statement

There is no switch-case statement in python

```python
1  print("Please choose an option")
2  print("1: To show balance")
3  print("2: To get cache")
4  print("e: To exit and get your cart !")
5  user_choice = input()
6  if user_choice == '1':
7      print("Your Balance is 2500 $")
8
9  elif user_choice == '2':
10     # Here we should call a method that check input amount that user want to withdraw
11     # Here we should call a method that withdraw the cache
12     print("Please take your cache")
13
14 elif user_choice == 'e':
15     # Here we should
16     print("Please take your cart")
17 else:
18     print("Your input is invalid")
```

Lets try!

Ali Samanipour
Samani.Academy@gmail.com

Module Roadmap

1   The If Statement

2   The If-else Statement

3   Nested Decision Structures

4   The If-elif-else Statement

5   Logical Operators

Logical Operators

The logical and operator and the logical or operator allow you to
Connect multiple Boolean expressions to create a compound expression

The logical not operator reverses the truth of a Boolean expression

| **and** | **or** | **not** |

Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	x > 1 and x < 10
or	Returns True if one of the statements is true	x < 1 or x > 10
not	Reverse the result, returns False if the result is true	not (x > 5)

Ali Samanipour
Samani.Academy@gmail.com

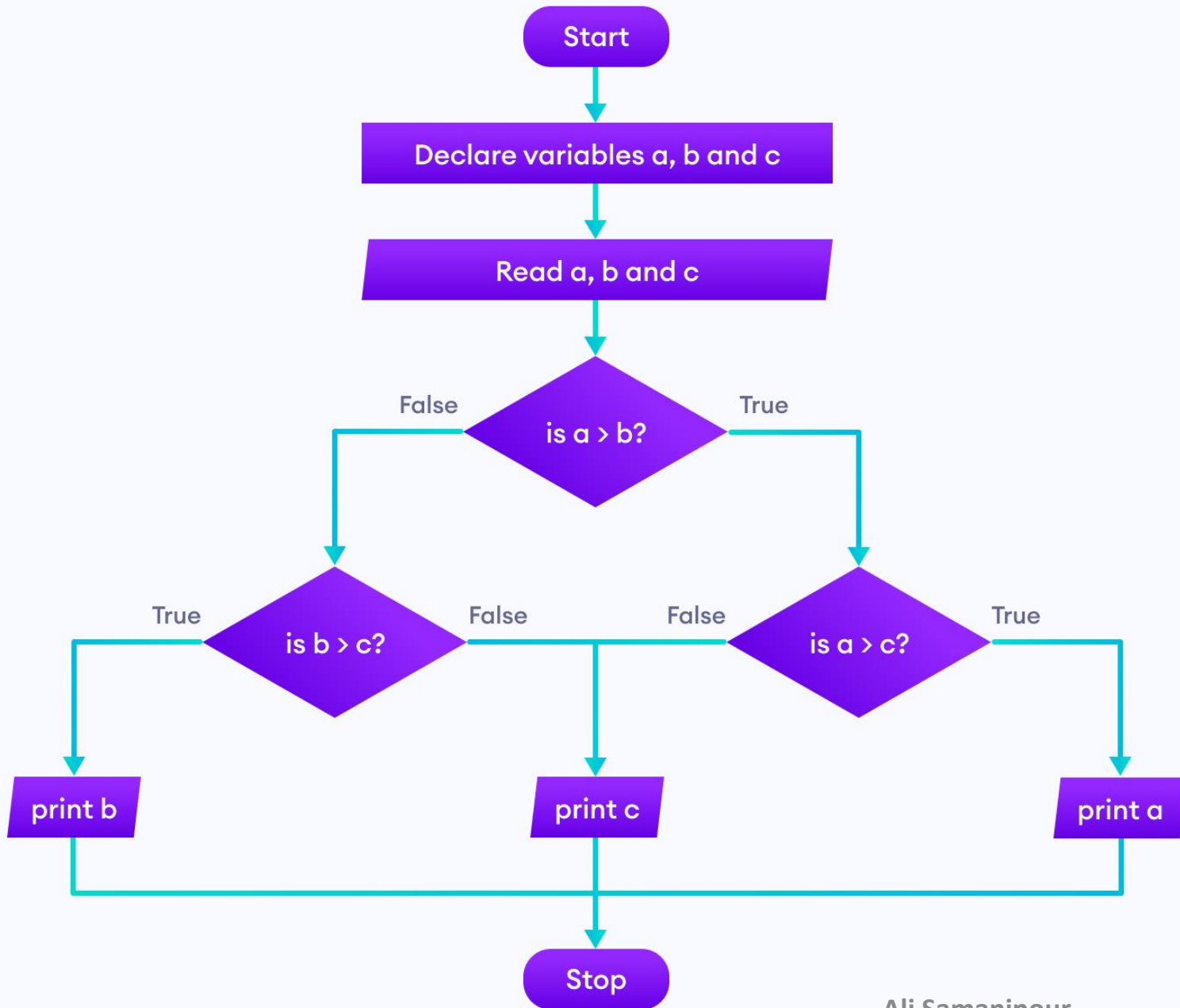# A decision structure, Grouping Boolean Expersions (Code Example)

```python
1  # This program determines whether a bank customer
2  # qualifies for a loan.
3  MIN_SALARY = 30000.0 # The minimum annual salary
4  MIN_YEARS = 2 # The minimum years on the job
5  # Get the customer's annual salary.
6  salary = float(input('Enter your annual salary: '))
7  # Get the number of years on the current job.
8  years_on_job = int(input('Enter the number of ' +'years employed: '))
9  # Determine whether the customer qualifies.
10 if salary >= MIN_SALARY and years_on_job >= MIN_YEARS:
11     print('You qualify for the loan.')
12 else:
13     print('You do not qualify for this loan.')
```

Truth Tables

Expression	Value of the Expression
true and false	false
false and true	false
false and false	false
true and true	true

Expression	Value of the Expression
true or false	true
false or true	true
false or false	false
true or true	true

Expression	Value of the Expression
not true	false
not false	true

Ali Samanipour
Samani.Academy@gmail.com

On Class Exercise! (Finding Maximum)

Ali Samanipour
Samani.Academy@gmail.com

31

```python
1   a =9
2   b =6
3   c =5
4   if(a > b):
5       if(a > c):
6           print(a)
7       else:
8           print(c)
9   else:
10      if(b > c):
11          print(b)
12      else:
13          print(c)
14
```

On Class Exercise! (Finding Maximum Solution1)

On Class Exercise! (Finding Maximum Solution1)

```
1   a =9
2   b =6
3   c =5
4   if(a > b):
5       if(a > c):
6           print(a)
7       else:
8           print(c)
9   else:
10      if(b > c):
11          print(b)
12      else:
13          print(c)
14
```

Try grouping these conditions using and oprators

```python
1  if ((a > b) and (a > c)):
2      print(a)
3  if ((a > b) and not(a > c)):
4      print(c)
5  if (not(a > b) and (b > c)):
6      print(b)
7  if (not(a > b) and not(b > c)):
8      print(c)
```

On Class Exercise! (Finding Maximum Solution2)

Ali Samanipour
Samani.Academy@gmail.com

```
1  #Printing Turth Table
2
3  print (True and True)
4  print (True and False)
5  print (False and False)
6
7  print (True or True)
8  print (True or False)
9  print (False or False)
10
11 print (not True)
12 print (not False)
```

Lets try!

Ali Samanipour
Samani.Academy@gmail.com

Appendix A: Assignment Operators

Operator	Example	Same as
=	x = 7	x = 7
+=	x += 7	x = x+7
-=	x -= 7	x = x-7
*=	x *= 7	x = x*7
/=	x /= 7	x = x/7
//=	x //= 7	x = x//7

Appendix A: Assignment Operators

Operator	Example	Same as
%=	x %= 7	x = x % 7
**=	x **= 7	x = x**7
&=	x &= 7	x = x&7
|=	x |= 7	x = x|7
^=	x ^= 7	x = x^7
>>=	x >>= 7	x = x>>7

Ali Samanipour
Samani.Academy@gmail.com

Appendix B: Membership Operators

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

Ali Samanipour
Samani.Academy@gmail.com

Appendix D: Identity Operators

Operator	Description	Example
is	Returns True if both variables are the same object	x is y
Is not	Returns True if both variables are not the same object	x is not y

Ali Samanipour
Samani.Academy@gmail.com

Appendix C: Bitwise Operators

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
|	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
>>	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
<<	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

Ali Samanipour
Samani.Academy@gmail.com