SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Lecture 4

Planning a Procedure using Pseudocode

Selection: If-elif-else

KNOWING

Planning a Procedure using Pseudocode

- Using pseudocode to plan a procedure
  - Pseudocode: Short phrases used to describe the steps a procedure must take to accomplish its goal.
  - Travel directions are a type of pseudocode

Planning a Procedure using Pseudocode

- Example:

    1. Input 2 numbers
    2. Calculate Total and Average
    3. Display 2 numbers, Total, and Average

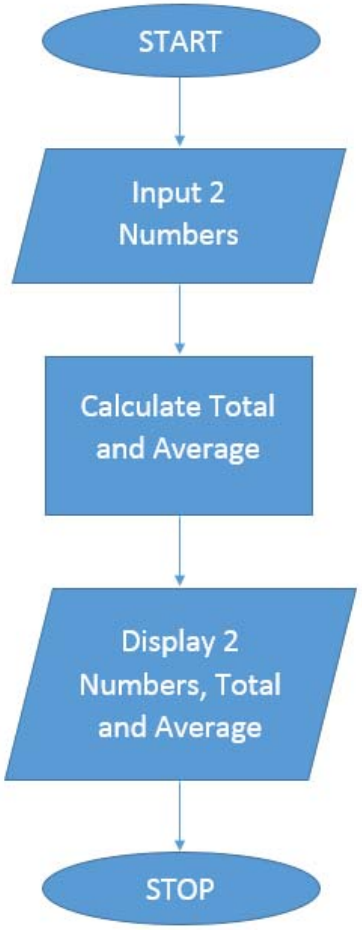
Planning a Procedure using a Flowchart

- Using a flowchart to plan a procedure.
- A flowchart shows program logic using standardized symbols
  - Oval: Start/Stop symbol
  - Rectangle: Process symbol; represents a task
  - Parallelogram: Input/Output symbol
  - Flowlines: connect the symbols
- Flowcharts depict same logic as pseudocode.

KNOW ING

Planning a Procedure using a Flowchart

- Example:



Making Decisions in a Program (revisit)

- Three basic control structures
  - Sequence
  - Selection
  - Repetition
- All procedures in an application are written using one of more of these structures.

KNOW ING

Making Decisions in a Program (revisit)

- Selection structure
  - Chooses one of two paths based on condition
  - Also called a decision structure
- Example:
  - If employee works over 40 hours, add overtime pay
- Condition
  - Decision expression evaluating to true or false

KNOW ING

Making Decisions in a Program (revisit)

- Single-alternative selection structure
  - Tasks performed only when condition is true
- Dual-alternative selection structure
  - One set of tasks performed if condition is true
    - Called true path
  - Different set of tasks performed if condition is false
    - Called false path

KNOW ING

Making Decisions in a Program (revisit)

Example 1: Single-alternative selection structure

condition

If it is raining

wear a rain coat
bring an umbrella } True path

Example 2: Dual-alternative selection structure

condition

If you have a test tomorrow

study tonight → true path

Else:

watch a movie → false path

KNOW ING

Python If..elif..else

- The if-elif-else statement is used to conditionally execute a statement or block of statements.
- Conditions can be true or false, execute one thing when the condition is true, something else when the condition is false.

KNOW ING

Python If..elif..else

- Contents:
  - If statement
  - If .. Else statement
  - If .. Elif .. Else statement
  - Nested if .. Else
  - Use the AND operator in an if statement
  - Use the IN operator in an if statement
  - Write an if-else in a single line of code
  - Define a negative if

KNOW ING

if statement

- The Python if statement is same as it is with other programming languages. It executes a set of statements conditionally, based on the value of a logical expression.

if statement

- Syntax:

```
If expression:
    statement_1
    statement_2
    ……
```

- In the above case, expression specifies the conditions which are based on Boolean expression.

if statement

- When a Boolean expression is evaluated it produces either a value of true or false.
- If the expression evaluates true the same amount of indented statement(s) following if will be executed.
- This group of statement(s) is called a block.

KNOW ING

if .. else statement

- In Python if .. else statement, if has two blocks, one following the expression and other following the else clause.
- Syntax:

```
If expression:
        statement_1
        statement_2
        ......
        else:
        statement_3
        statement_4
        ......
```

if .. else statement

- If the expression evaluates to true the same amount of indented statements(s) following if will be executed and if the expression evaluates to false the same amount of indented statements(s) following else will be executed.

if .. else statement

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
a=10
if(a>10):
    print("Value of a is greather than 10")
else:
    print("Value of a is 10")
```

Ln: 5 Col: 25

```
>>>
 RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_state
ment.py
Value of a is 10
>>>
```

KNOW ING

if .. elif .. else .. statement

- Sometimes a situation arises when there are several conditions.
- To handle the situation Python allows adding number of elif clause after an if and before an else clause.

if .. elif .. else .. statement

- Syntax:

```
If expression1:
    statement_1
    statement_2
    ……
    elif expression2:
    statement_3
    statement_4
    ……
    elif expression3:
    statement_5
    statement_6
    else:
    ……
    statement_7
    statement_8
```

if .. elif .. else .. statement

- Python evaluates each expression (i.e. the condition) one by one and if a true condition is found the statement(s) block under the expression will be executed.
- If no true condition is found the statement(s) block under else will be executed.

if .. elif .. else .. statement

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
var1 = 1+2j
if(type(var1)==int):
    print("Type of the variable is Integer")
elif(type(var1) == float):
    print("Type of the variable is Float")
elif(type(var1) == complex):
    print("Type of the variable is Complex")
elif(type(var1) == bool):
    print("Type of the variable is Bool")
elif(type(var1) == str):
    print("Type of the variable is String")
elif(type(var1) == tuple):
    print("Type of the variable is Tuple")
elif(type(var1) == dict):
    print("Type of the variable is Dictionaries")
elif(type(var1) == list):
    print("Type of the variable is List")
else:
    print("Type of the variable is Unknow")
```

Ln: 18 Col: 5

```
 RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_state
ment.py
Type of the variable is Complex
>>>
```

Ln: 12 Col: 4

KNOW ING

Nested if .. else statement

- In general nested if-else statement is used when we want to check more than one conditions.
- Conditions are executed from top to bottom and check each condition whether it evaluates to true or not.
- If a true condition is found the statement(s) block associated with the condition executes others it goes to next condition.

Nested if .. else statement

- Syntax:

```
If expression1:
  if expression2:
    statement_1
    statement_2
    ……
    else:
    statement_3
    statement_4
    ……
    else:
    statement_5
    statement_6
```

Nested if .. else statement

- Syntax expression1 is checked first, if it evaluates to true then the program control goes to next if-else part otherwise it goes to the last else statement and executes statement_5, statement_6 etc.
- Within the if-else if expression2 evaluates true then statement_1, statement_2 will execute, otherwise statement_3, statement_4 will execute.

KNOW ING

Nested if .. else statement

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
age=38
if(age>=11):
    print("You are eligible to see the Football match.")
    if(age<=20 or age>=60):
        print("Ticket price is $12")
    else:
        print("Ticket price is $20")
else:
    print("You're not eligible to buy a ticket.")
```

Ln: 11 Col: 0

```
ment.py
You are eligible to see the Football match.
Ticket price is $20
>>>
```

Ln: 16 Col: 4

KNOW ING

Use the and operator in an if statement

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
#create two boolean objects
x = False
y = True

#The validation will be True only if all the expressions generate a value True
if x and y:
    print('Both x and y are true')
else:
    print('x is False or y is False or both x and y are False')
```

Ln: 3 Col: 8

```
 RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_state
ment.py
x is False or y is False or both x and y are False
>>>
```

Ln: 25 Col: 4

KNOW ING

Use the in operator in an if statement

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
#create a string
s = 'JQuery'

#create a list
l = ['JavaScript', 'JQuery', 'ZinoUI']

#in operator is used to replace various expressions that use the or operator

if s in l:
    print(s + ' Tutorial')

#Alternate if statement with or operator

if s=='JavaSript' or s=='JQuery' or s=='ZinoUI':
    print(s + ' Tutorial')
```

Ln: 10 Col: 14

```
 RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_state
ment.py
JQuery Tutorial
JQuery Tutorial
>>>
```

Ln: 29 Col: 4

KNOW ING

Write an if-else in a single line of code

If_Else_statement.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_statem...

File Edit Format Run Options Window Help

```
#create an integer
n = 150
print(n)

#if n is greater than 500, n is multiplied by 7, otherwise n is divided by 7
result = n * 7 if n > 500 else n / 7
print(result)
```

Ln: 8 Col: 0

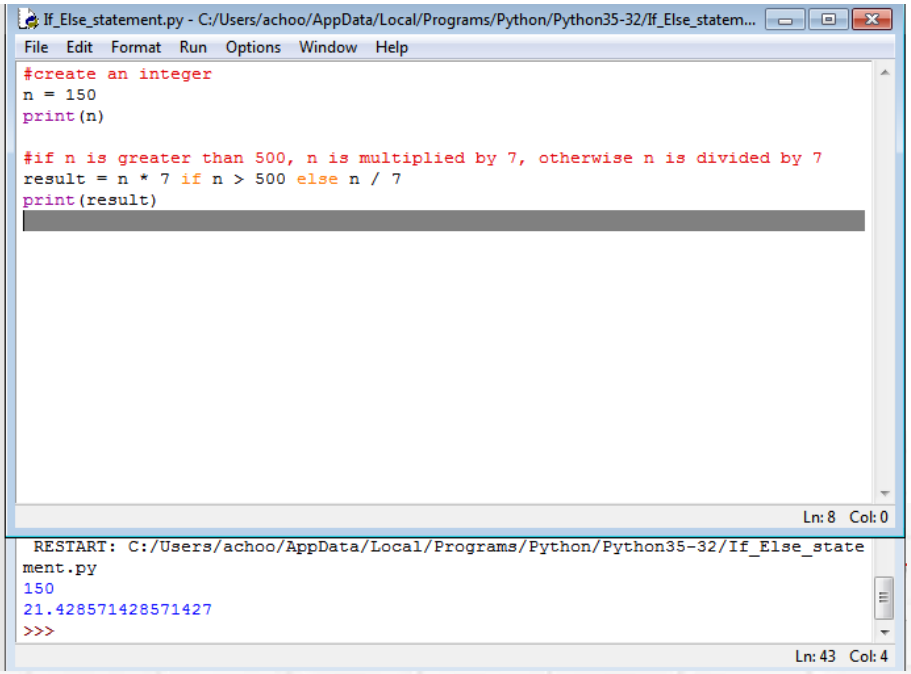
```
 RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/If_Else_state
ment.py
150
21.428571428571427
>>>
```

Ln: 43 Col: 4

KNOW ING

Define a negative (-) if

- If a condition is true the not operator is used to reverse the logical state, then logical not operator will make it false.



Generate Random numbers

- Using the randint() function
  - This function requires 2 integer argument values and returns a random integer between those 2 values, which may include either of the argument values.
- Using the randrange() function
  - This function returns a random integer from 0 however it does not reaching the last number (indicate in the range), thus +1 is used to reach the last number.

Generate Random numbers

Chap2.py - C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/Chap2.py (3.5.3)

File Edit Format Run Options Window Help

```
#Demonstrates random number generation

import random

#generate random numbers 1-6
die1 = random.randint(1,6)
die2 = random.randrange(6)+1

total = die1 + die2

print('You rolled a', die1, 'and a', die2, 'for a total of', total)
input('\n\nPress the enter key to exit.')
```

Ln: 1 Col: 38

```
= RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/Chap2.py =
You rolled a 5 and a 2 for a total of 7


Press the enter key to exit.
>>>
= RESTART: C:/Users/achoo/AppData/Local/Programs/Python/Python35-32/Chap2.py =
You rolled a 3 and a 5 for a total of 8


Press the enter key to exit.
>>>
```

Ln: 88 Col: 4

KNOW ING

Comparison Operators

Operator	Meaning	Sample Condition	Evaluates To
==	Equal to	5 == 5	True
!=	Not equal to	5 != 8	True
>	Greater than	3 > 10	False
<	Less than	5 < 8	True
>=	Greater than or equal to	5 >= 10	False
<=	Less than or equal to	5 <= 10	True

KNOW ING