



# Computer Engineering

## วิศวกรรมคอมพิวเตอร์



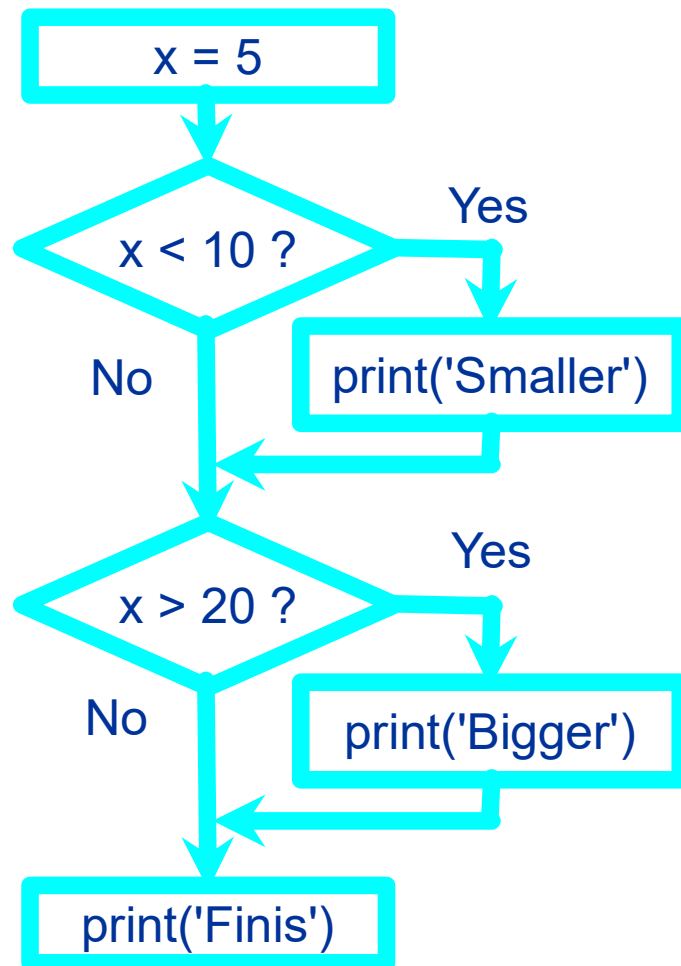
### บทที่ 3

## Conditional Execution

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

# Conditional Steps

01006012 Computer Programming



Program:

```

x = 5
if x < 10:
    print('Smaller')
if x > 20:
    print('Bigger')

print('Finis')
  
```

Output:

Smaller  
Finis

# Comparison Operators



01006012 Computer Programming

- Boolean expressions ask a question and produce a Yes or No result which we use to control program flow
- Boolean expressions using comparison operators evaluate to True / False or Yes / No

Comparison operators look at variables but do not change the variables

| Python | Meaning                  |
|--------|--------------------------|
| <      | Less than                |
| <=     | Less than or Equal to    |
| ==     | Equal to                 |
| >=     | Greater than or Equal to |
| >      | Greater than             |
| !=     | Not equal                |

Remember: “=” is used for assignment.

[http://en.wikipedia.org/wiki/George\\_Boole](http://en.wikipedia.org/wiki/George_Boole)

# การใช้งานเครื่องหมายการเปรียบเทียบ

01006012 Computer Programming

| การเปรียบเทียบ | ผลที่ได้ |
|----------------|----------|
| $7 == 9$       | False    |
| $7 != 9$       | True     |
| $8 > 8$        | False    |
| $8 >= 8$       | True     |
| $(10+9) < 7$   | False    |
| $4 <= 3$       | False    |

| การเปรียบเทียบ | ผลที่ได้ |
|----------------|----------|
| $22 == 22$     | True     |
| $(3+5) != 8$   | False    |
| $9 > 7$        | True     |
| $7 >= 9$       | False    |
| $7 < (10+9)$   | True     |
| $3 <= 4$       | True     |

ไม่ควรใช้เครื่องหมายเท่ากับ  $==$  หรือไม่เท่ากับ  $!=$  สำหรับข้อมูลทศนิยม



# Comparison Operators

01006012 Computer Programming

```
x = 5
if x == 5 :
    print('Equals 5')
if x > 4 :
    print('Greater than 4')
if x >= 5 :
    print('Greater than or Equals 5')
if x < 6 : print('Less than 6')
if x <= 5 :
    print('Less than or Equals 5')
if x != 6 :
    print('Not equal 6')
```

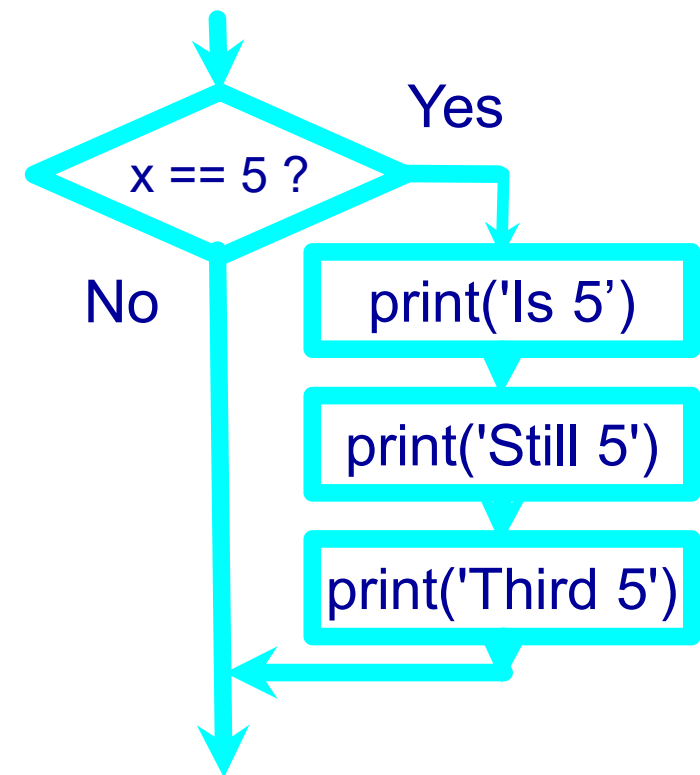
Equals 5  
Greater than 4  
Greater than or Equals 5  
Less than 6  
Less than or Equals 5  
Not equal 6

# One-Way Decisions

01006012 Computer Programming

```
x = 5
print('Before 5')
if x == 5 :
    print('Is 5')
    print('Is Still 5')
    print('Third 5')
print('Afterwards 5')
print('Before 6')
if x == 6 :
    print('Is 6')
    print('Is Still 6')
    print('Third 6')
print('Afterwards 6')
```

Before 5  
Is 5  
Is Still 5  
Third 5  
Afterwards 5  
Before 6  
Afterwards 6



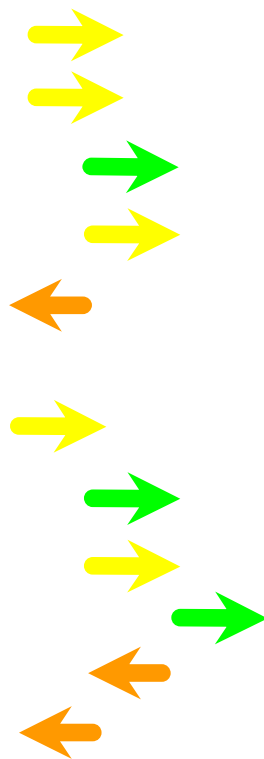
# Indentation



01006012 Computer Programming

- Increase indent after an if statement or for statement (after : )
- Maintain indent to indicate the scope of the block (which lines are affected by the if/for)
- Reduce indent back to the level of the if statement or for statement to indicate the end of the block
- Blank lines are ignored - they do not affect indentation
- Comments on a line by themselves are ignored with regard to indentation
- Avoid using tab
- 4 spacs is reccommend

increase / maintain after if or for  
decrease to indicate end of block



```
x = 5
if x > 2 :
    print('Bigger than 2')
    print('Still bigger')
print('Done with 2')

for i in range(5) :
    print(i)
    if i > 2 :
        print('Bigger than 2')
    print('Done with i', i)
print('All Done')
```



# Think About begin/end Blocks



01006012 Computer Programming

```
x = 5
```

```
if x > 2 :  
    print('Bigger than 2')  
    print('Still bigger')  
print('Done with 2')
```

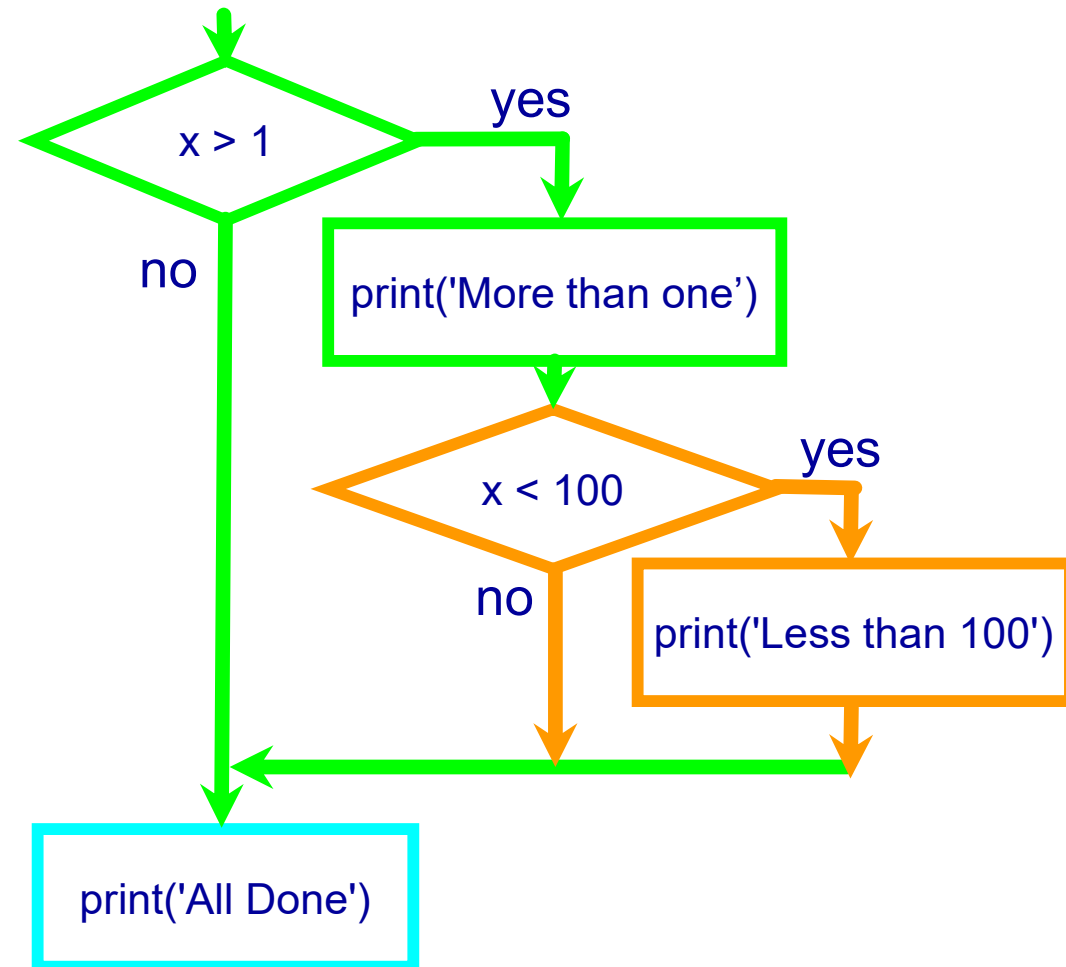
```
for i in range(5) :  
    print(i)  
    if i > 2 :  
        print('Bigger than 2')  
    print('Done with i', i)  
print('All Done')
```

# Nested Decisions



01006012 Computer Programming

```
x = 42
if x > 1 :
    print('More than one')
    if x < 100 :
        print('Less than 100')
print('All done')
```

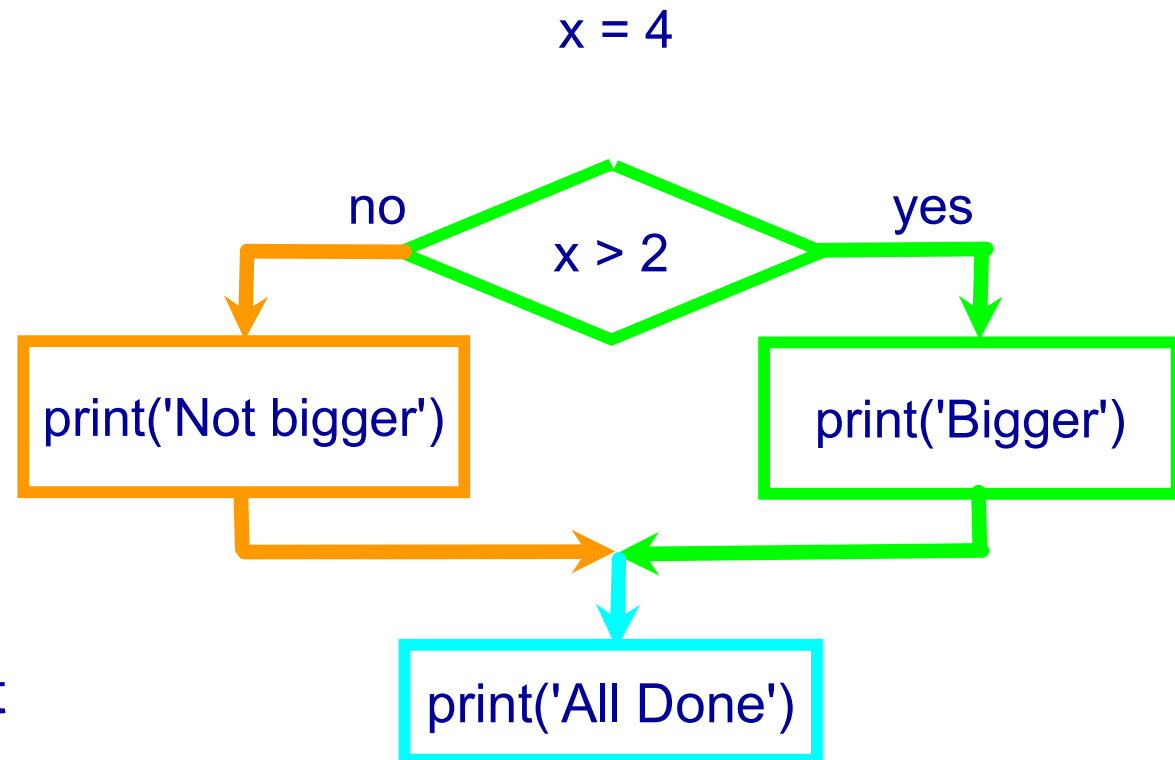


# Two-way Decisions

01006012 Computer Programming

Sometimes we want to do one thing if a logical expression is true and something else if the expression is false

It is like a fork in the road - we must choose one or the other path but not both

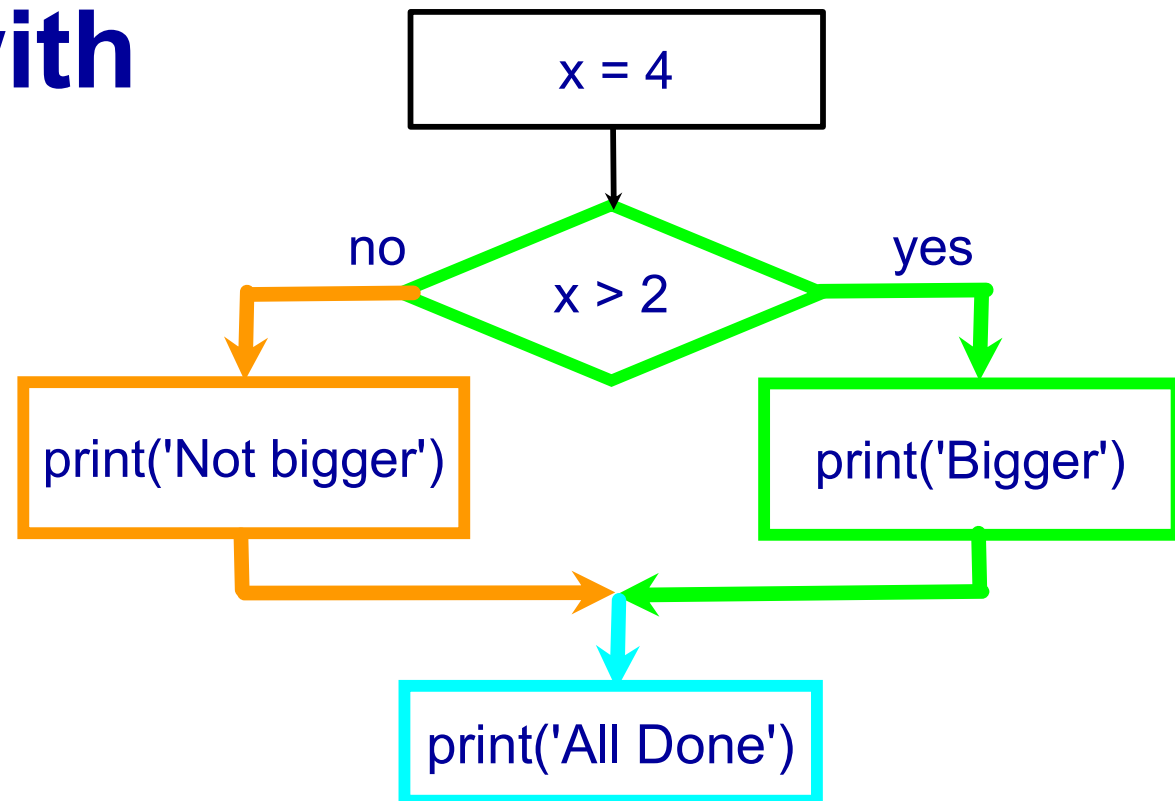


# Two-way Decisions with else:

```
x = 4
```

```
if x > 2 :
    print('Bigger')
else :
    print('Smaller')
```

```
print('All done')
```



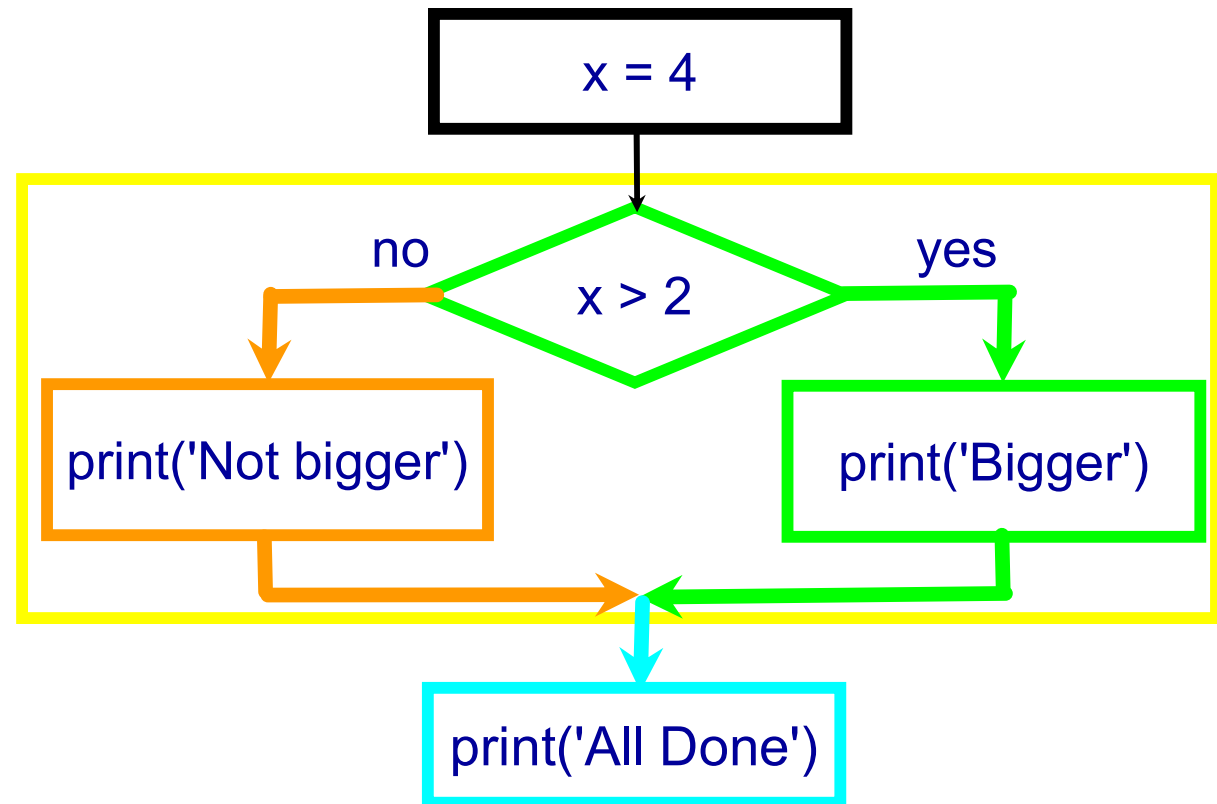
# Visualize Blocks

01006012 Computer Programming

`x = 4`

```
if x > 2 :
    print('Bigger')
else :
    print('Smaller')
```

`print('All done')`

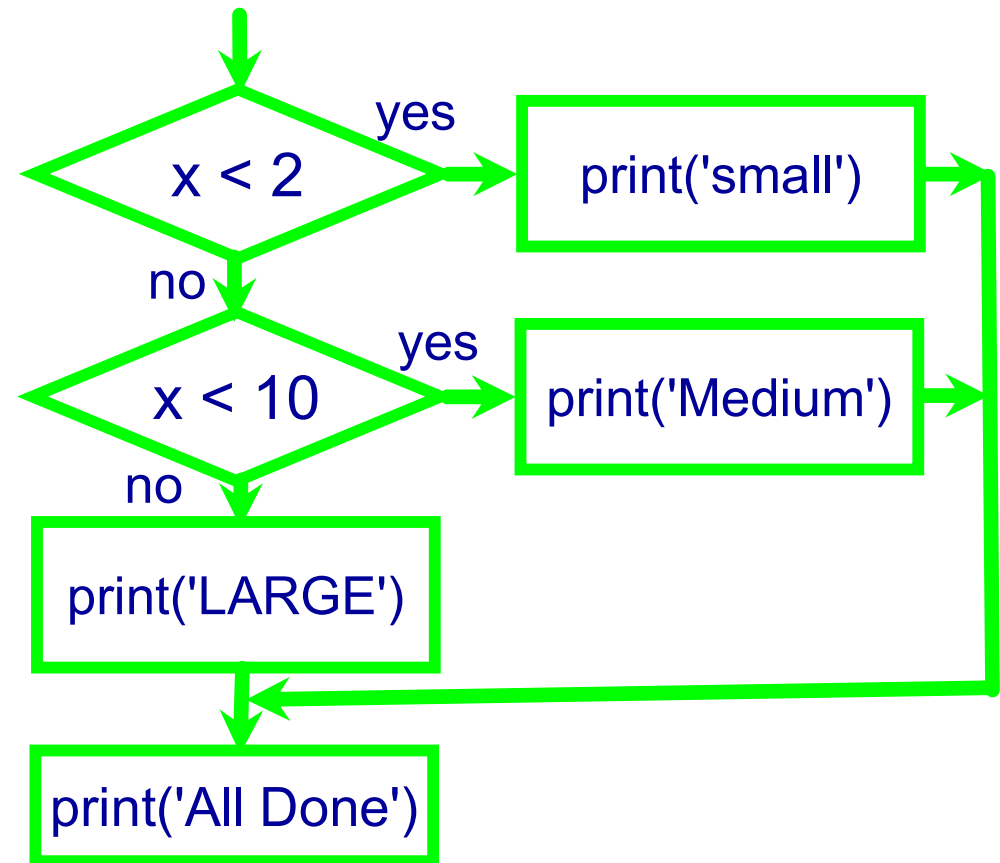




# More Conditional Structures...

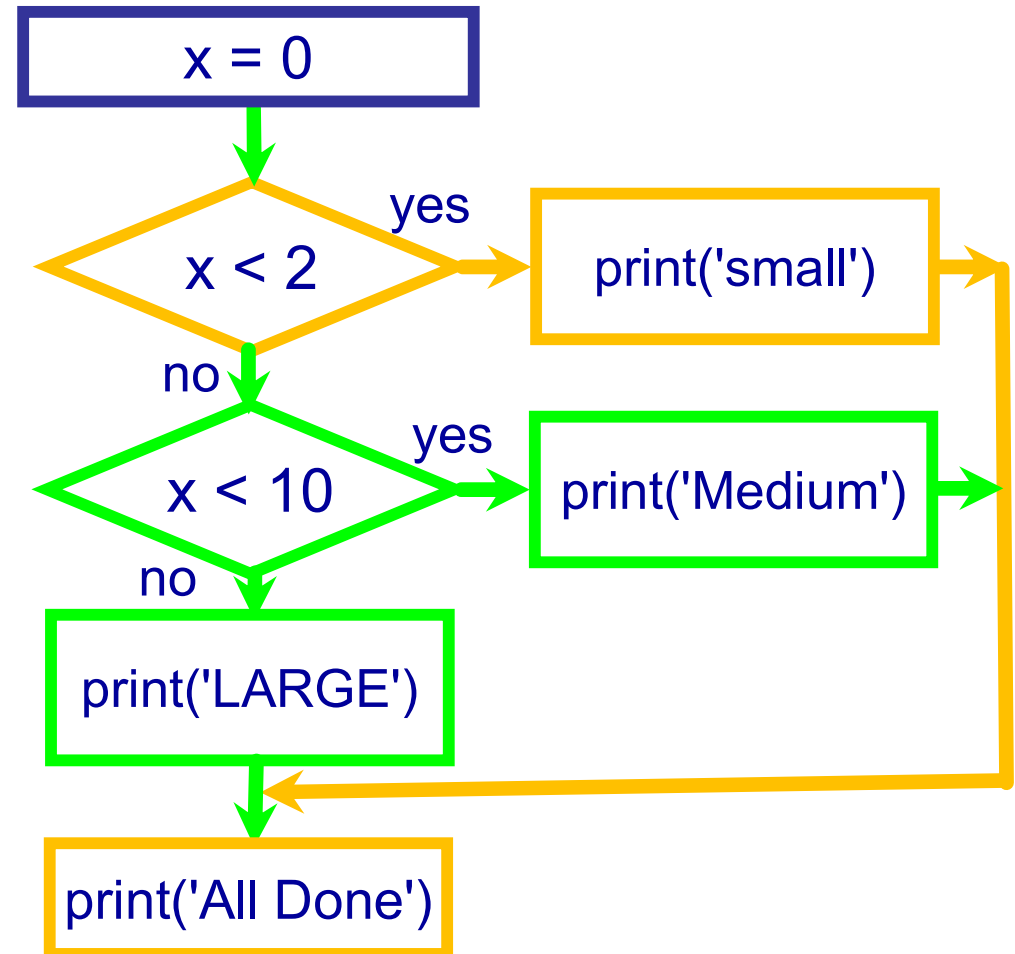
# Multi-way

```
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



# Multi-way

```
x = 0
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



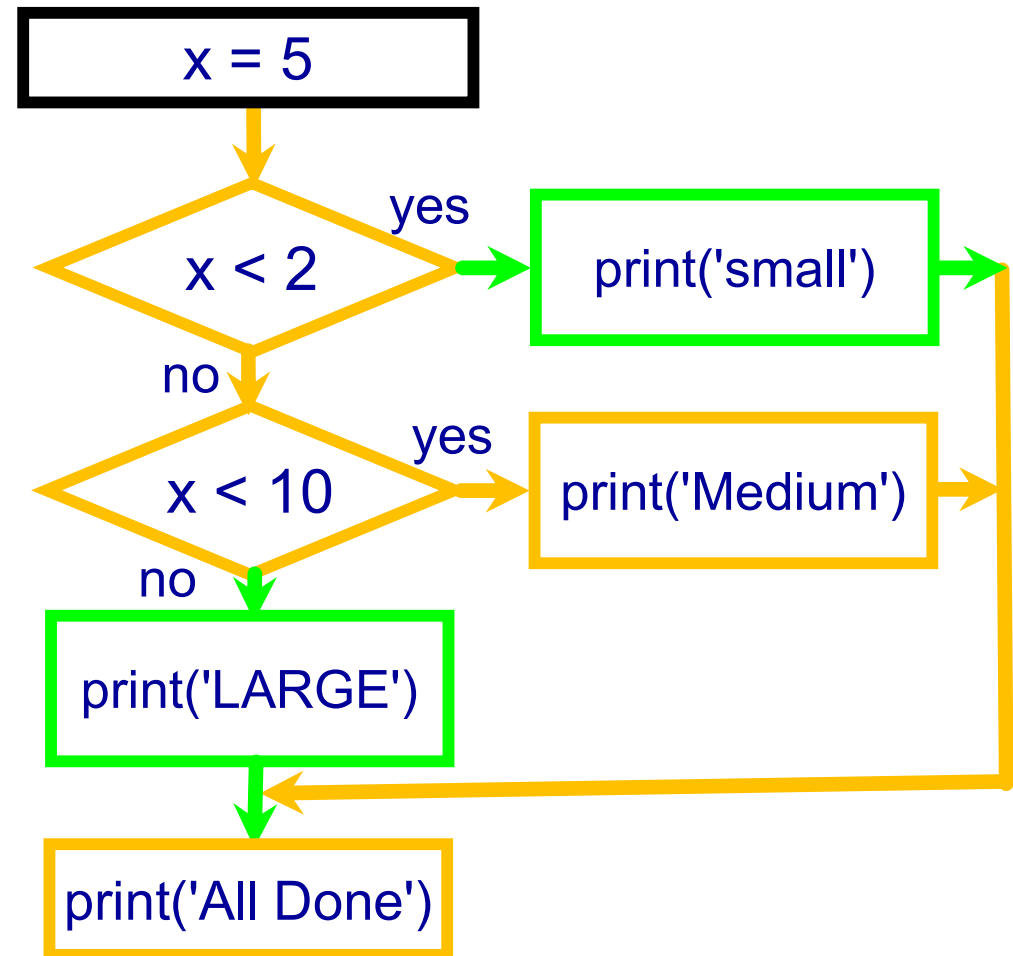


# Multi-way



01006012 Computer Programming

```
x = 5
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```

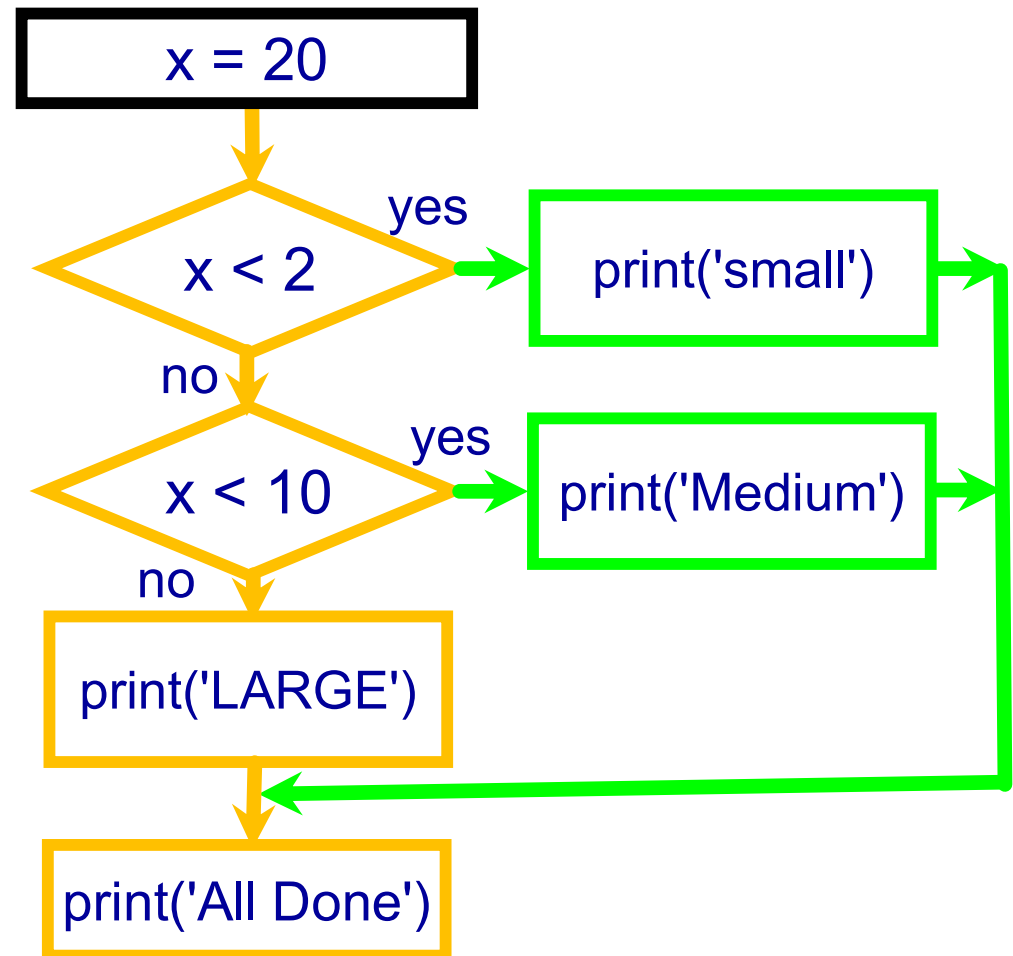


# Multi-way



01006012 Computer Programming

```
x = 20
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



# Multi-way



01006012 Computer Programming

```
# No Else
x = 5
if x < 2 :
    print('Small')
elif x < 10 :
    print('Medium')

print('All done')
```

```
if x < 2 :
    print('Small')
elif x < 10 :
    print('Medium')
elif x < 20 :
    print('Big')
elif x < 40 :
    print('Large')
elif x < 100:
    print('Huge')
else :
    print('Ginormous')
```



# Multi-way Puzzles

01006012 Computer Programming

Which will never print  
regardless of the value for x?

```
if x < 2 :  
    print('Below 2')  
elif x >= 2 :  
    print('Two or more')  
else :  
    print('Something else')
```

```
if x < 2 :  
    print('Below 2')  
elif x < 20 :  
    print('Below 20')  
elif x < 10 :  
    print('Below 10')  
else :  
    print('Something else')
```

# The try / except Structure



01006012 Computer Programming

You surround a dangerous section of code with try and except

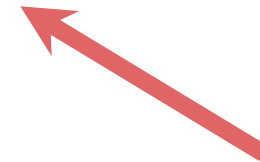
If the code in the try works - the except is skipped

If the code in the try fails - it jumps to the except section



```
$ type notry.py
astr = 'Hello Bob'
istr = int(astr)
print('First', istr)
astr = '123'
istr = int(astr)
print('Second', istr)
```

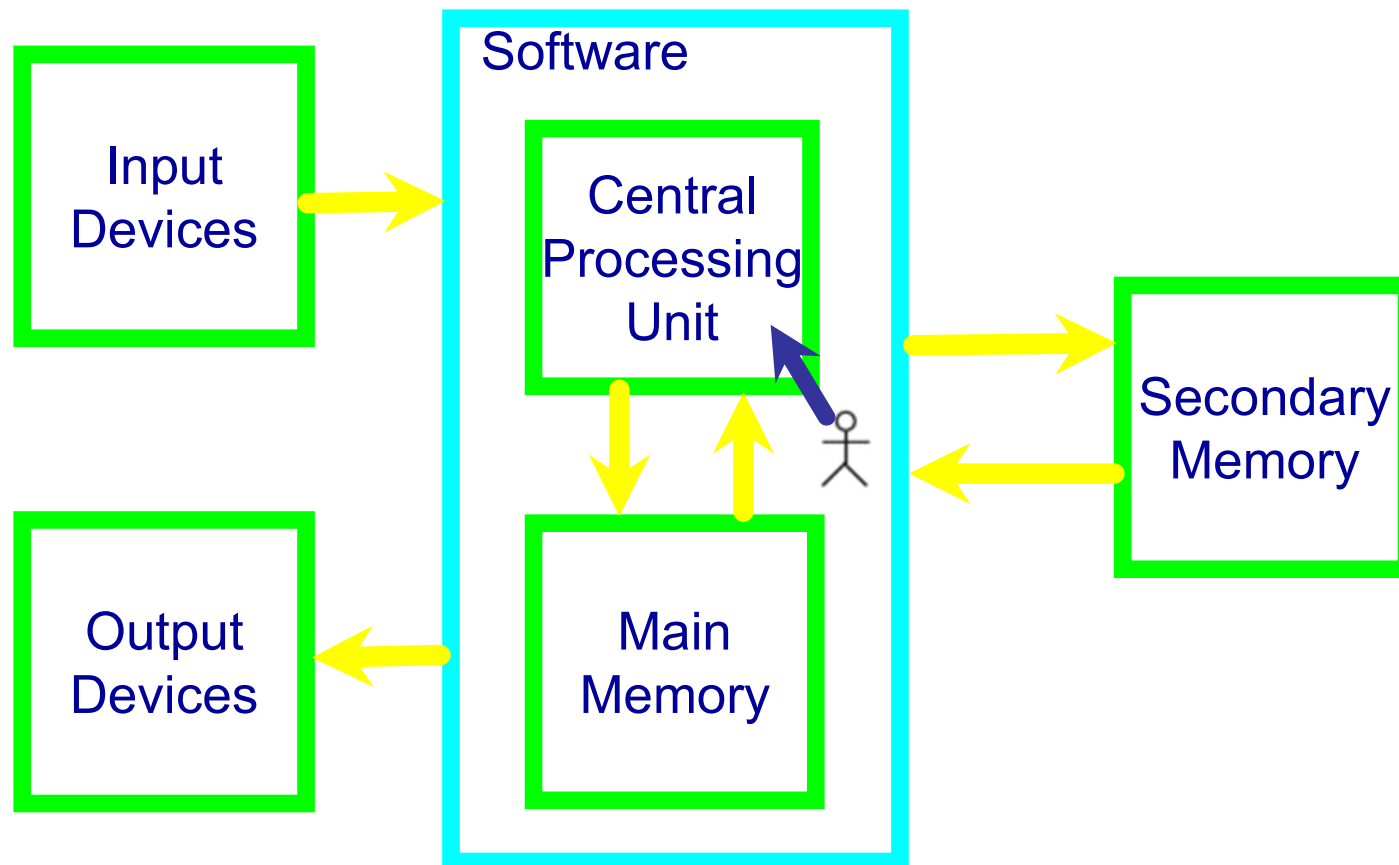
```
D:kt/ch03> python notry.py
Traceback (most recent call last):
File "notry.py", line 2, in <module>
istr = int(astr)ValueError: invalid literal
for int() with base 10: 'Hello Bob'
```

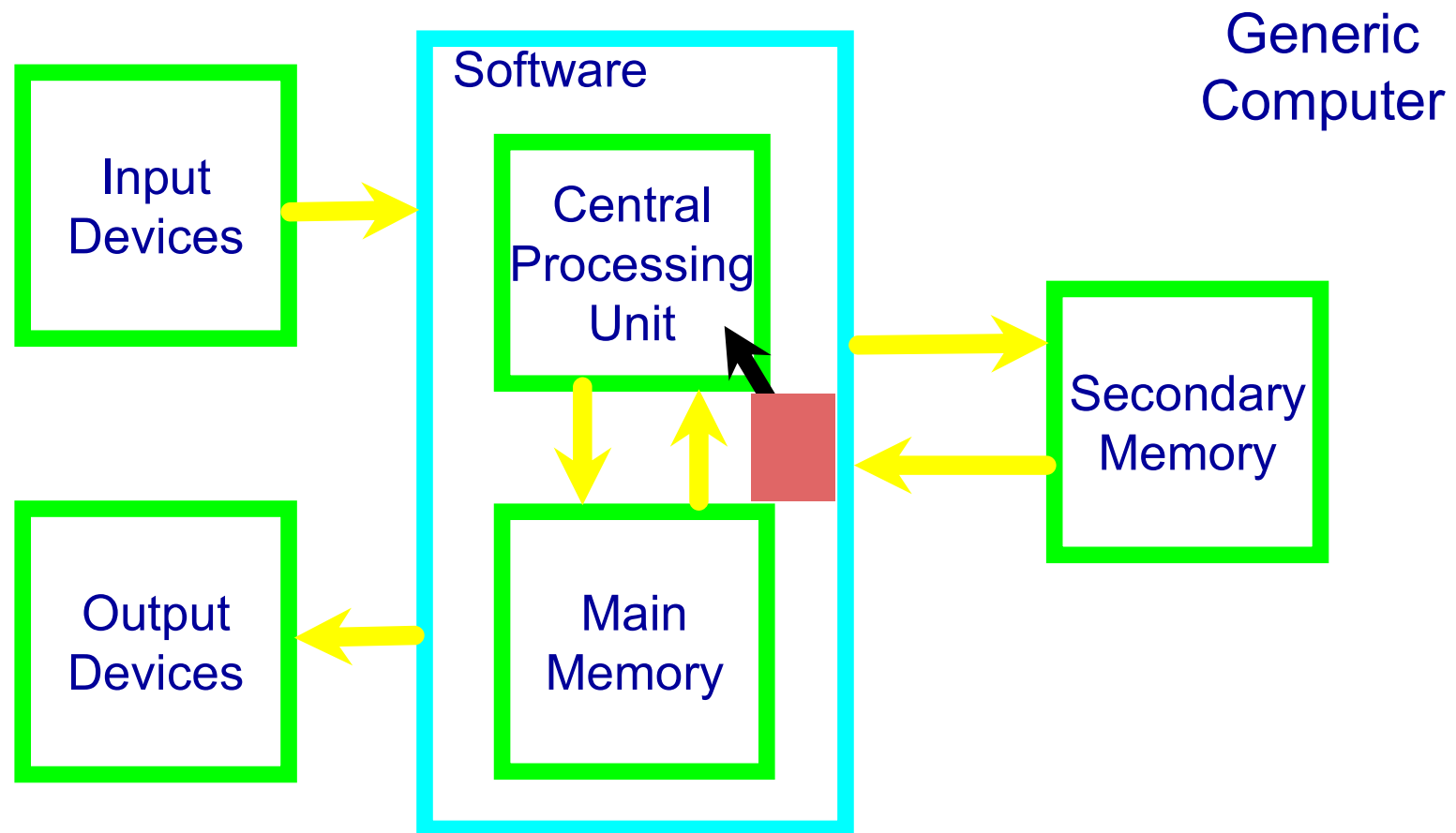


All  
Done

# Generic Computer

01006012 Computer Programming







```
astr = 'Hello Bob'
try:
    istr = int(astr)
except:
    istr = -1
```

```
print('First', istr)
```

```
astr = '123'
```

```
try:
    istr = int(astr)
except:
    istr = -1
```

```
print('Second', istr)
```

When the first conversion fails - it just drops into the except: clause and the program continues.

```
$ python tryexcept.py
First -1
Second 123
```

When the second conversion succeeds - it just skips the except: clause and the program continues.

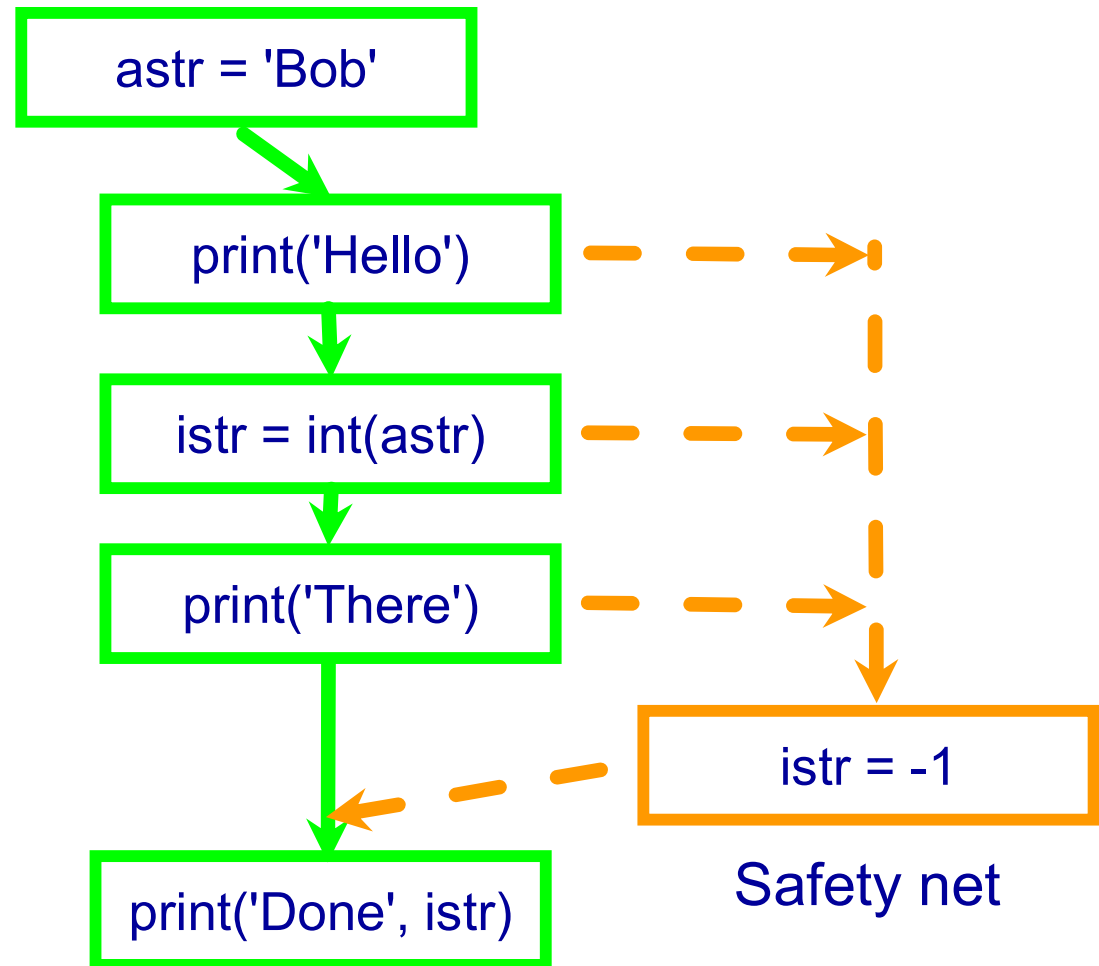
# try / except



01006012 Computer Programming

```
astr = 'Bob'
try:
    print('Hello')
    istr = int(astr)
    print('There')
except:
    istr = -1

print('Done', istr)
```





# Sample try / except

01006012 Computer Programming

```
rawstr = input('Enter a number:')
try:
    ival = int(rawstr)
except:
    ival = -1

if ival > 0 :
    print('Nice work')
else:
    print('Not a number')
```

```
$ python3 trynum.py
Enter a number:42
Nice work
$ python3 trynum.py
Enter a number:forty-two
Not a number
$
```

# ตัวดำเนินการทางตรรกศาสตร์ (Logical operators)



01006012 Computer Programming

| เครื่องหมาย | ความหมาย                  | ตัวอย่าง       |
|-------------|---------------------------|----------------|
| <b>and</b>  | และ (and)                 | <b>x and y</b> |
| <b>or</b>   | หรือ (or)                 | <b>x or y</b>  |
| <b>not</b>  | ไม่ หรือ ตรงกันข้าม (not) | <b>not x</b>   |

# การใช้งานเครื่องหมายทางตรรกศาสตร์

01006012 Computer Programming

| การดำเนินการ   | ผลที่ได้ |
|----------------|----------|
| <b>T and T</b> | <b>T</b> |
| <b>T and F</b> | <b>F</b> |
| <b>F and T</b> | <b>F</b> |
| <b>F and F</b> | <b>F</b> |

| การดำเนินการ  | ผลที่ได้ |
|---------------|----------|
| <b>T or T</b> | <b>T</b> |
| <b>T or F</b> | <b>T</b> |
| <b>F or T</b> | <b>T</b> |
| <b>F or F</b> | <b>F</b> |

| การดำเนินการ | ผลที่ได้ |
|--------------|----------|
| <b>not T</b> | <b>F</b> |
| <b>not F</b> | <b>T</b> |

# การใช้งานเครื่องหมายทางตรรกศาสตร์

01006012 Computer Programming

```
num1, num2, num3 = 10, 20, 30
```

```
num1 == num2
```

```
num1 > num2
```

```
(num1 < num2) and (num2 < num3)
```

```
(num1 > num2) or (num1 > num3)
```

```
(num1 > num2) or (num2 < num3)
```

## 4.2.1 การเปรียบเทียบ

01006012 Computer Programming

| เครื่องหมาย  | การเปรียบเทียบ      | ตัวอย่าง         |
|--------------|---------------------|------------------|
| <b>==</b>    | เท่ากับ             | <b>x == y</b>    |
| <b>!=</b>    | ไม่เท่ากับ          | <b>x != y</b>    |
| <b>&gt;</b>  | มากกว่า             | <b>x &gt; y</b>  |
| <b>&gt;=</b> | มากกว่าหรือเท่ากับ  | <b>x &gt;= y</b> |
| <b>&lt;</b>  | น้อยกว่า            | <b>x &lt; y</b>  |
| <b>&lt;=</b> | น้อยกว่าหรือเท่ากับ | <b>x &lt;= y</b> |

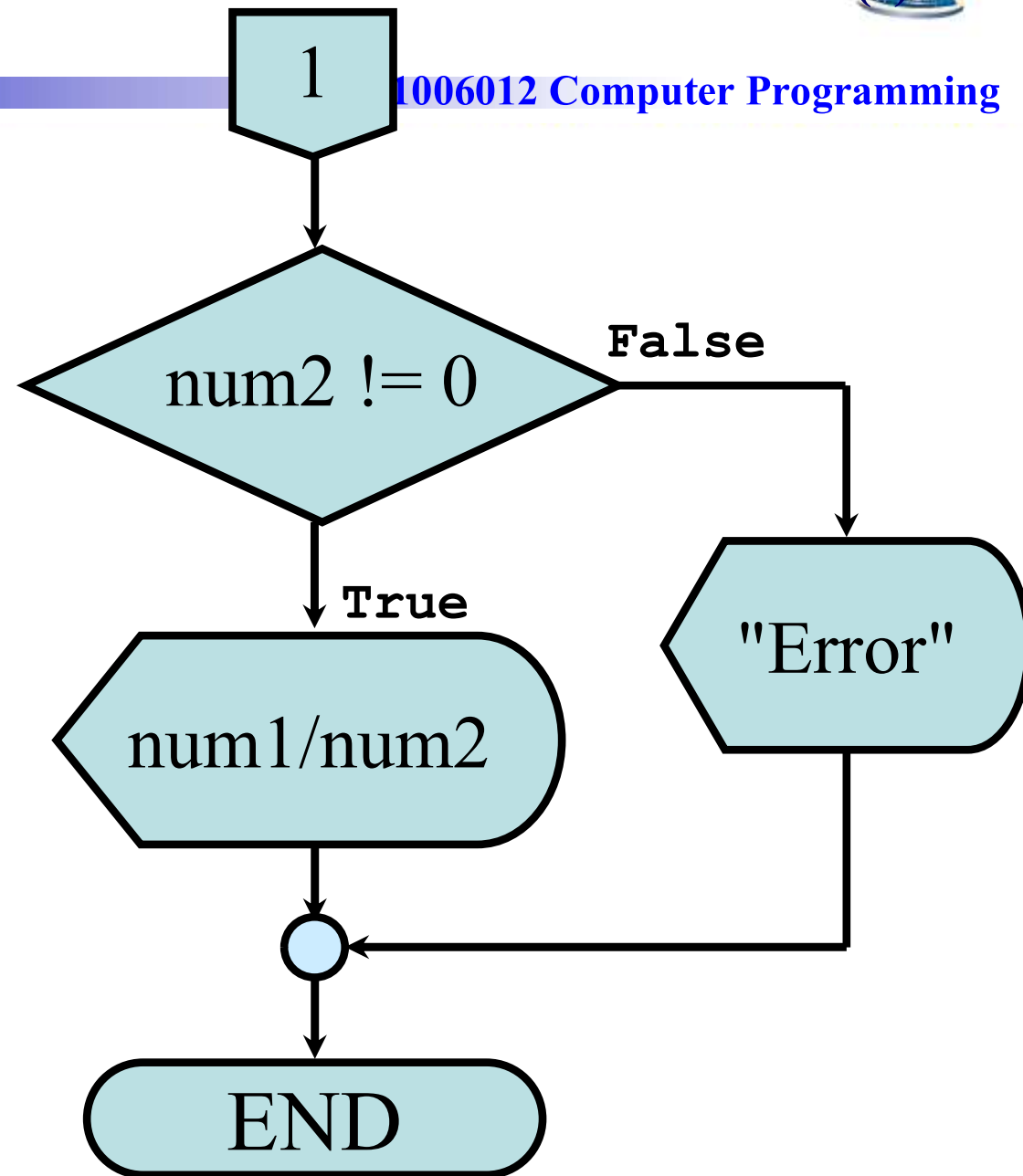
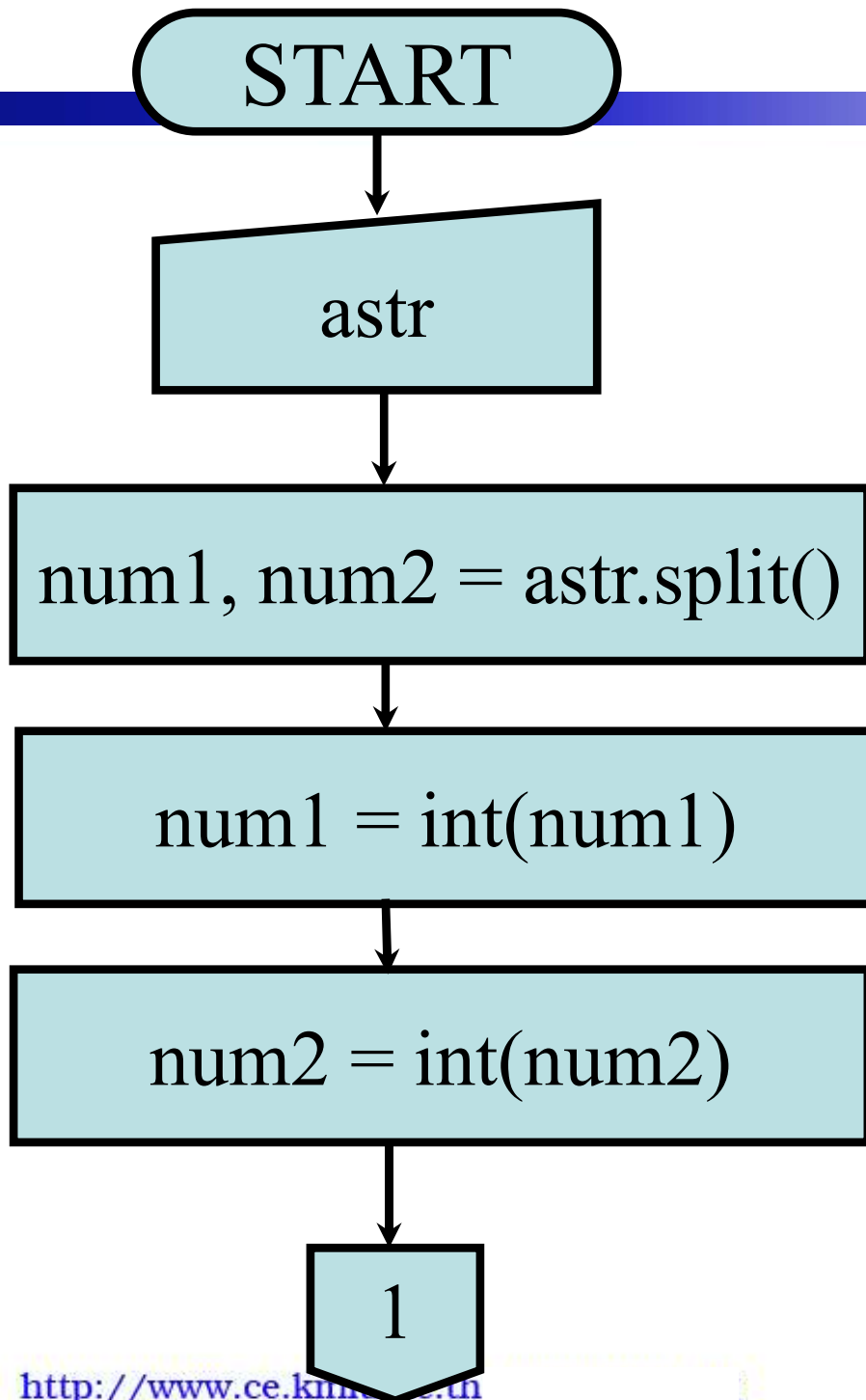
# โปรแกรม4.3 หารเลข 2 จำนวน

01006012 Computer Programming

จงเขียนผังงานและโปรแกรมหารเลข 2 จำนวน โดยโปรแกรม  
ต้องตรวจสอบได้ว่าตัวหารเป็น "0" หรือไม่

- Output Analysis
  - แสดงผลหารของเลข 2 จำนวน
  - แสดงผลว่าไม่สามารถหารได้เพราะตัวหารเป็นศูนย์
- Input Analysis
  - ตัวตั้ง และตัวหาร





# โปรแกรมหารเลข | if-else

01006012 Computer Programming

```
astr = input("Enter num1 num2 : ")
num1,num2 = astr.split()
num1 = int(num1)
num2 = int(num2)
if num2==0:
    print("Error divided by zero")
else:
    print(f"{num1} / {num2} = {num1/num2}")
```

```
Enter num1 num2 : 2 0
Error divided by zero
```

```
Enter num1 num2 : 3 5
3 / 5 = 0.6
```

# โปรแกรม ตรวจสอบเกรด A ถึง F

01006012 Computer Programming

จงเขียนผังงานและโปรแกรมสำหรับรับชื่อ และคะแนนวิชา  
Computer Programming เพื่อตรวจสอบว่านักศึกษาได้เกรด  
ระดับใด โดยใช้เกณฑ์ดังนี้

คะแนน 82 – 100 ได้เกรด A

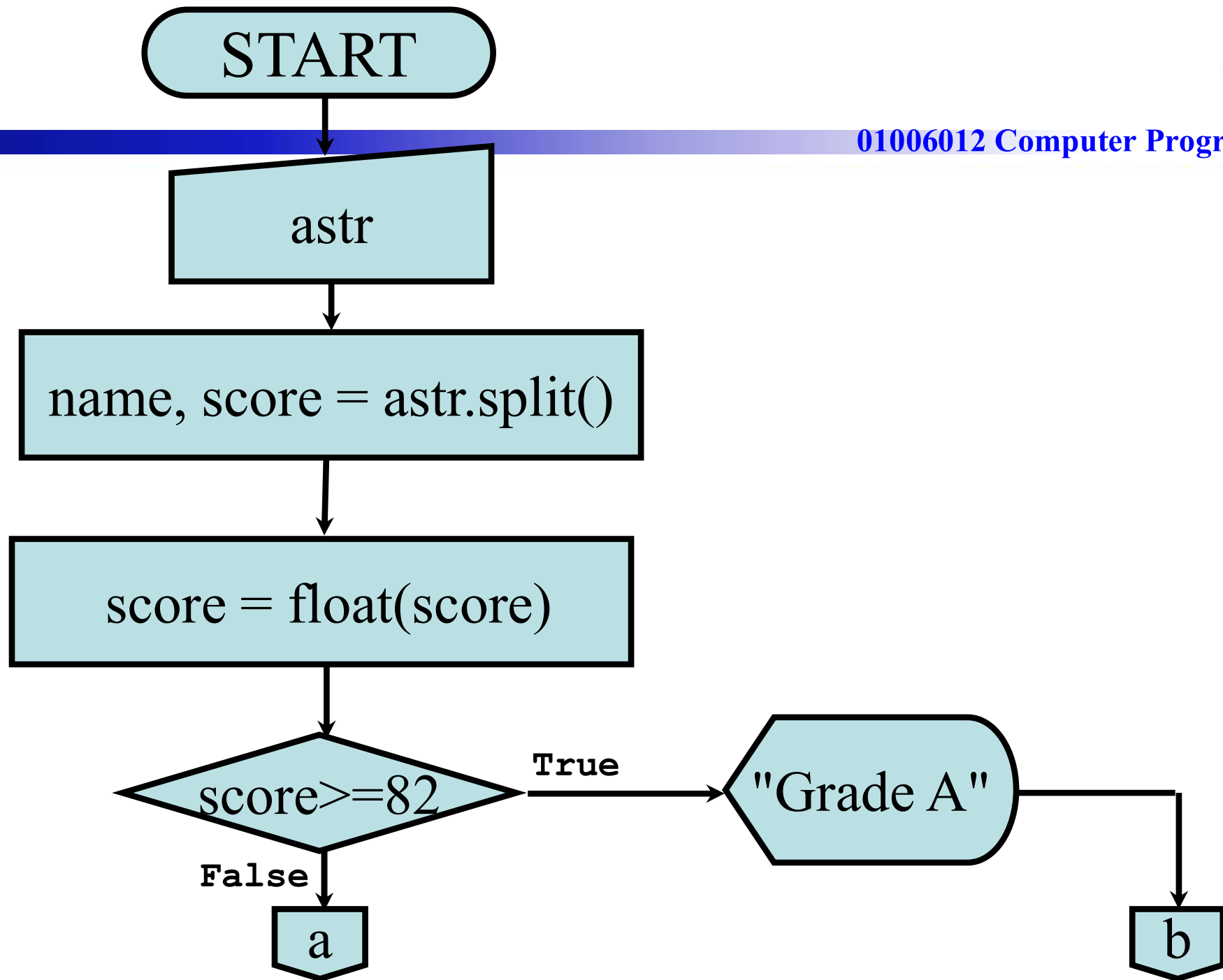
คะแนน 54 – 67.99 ได้เกรด C

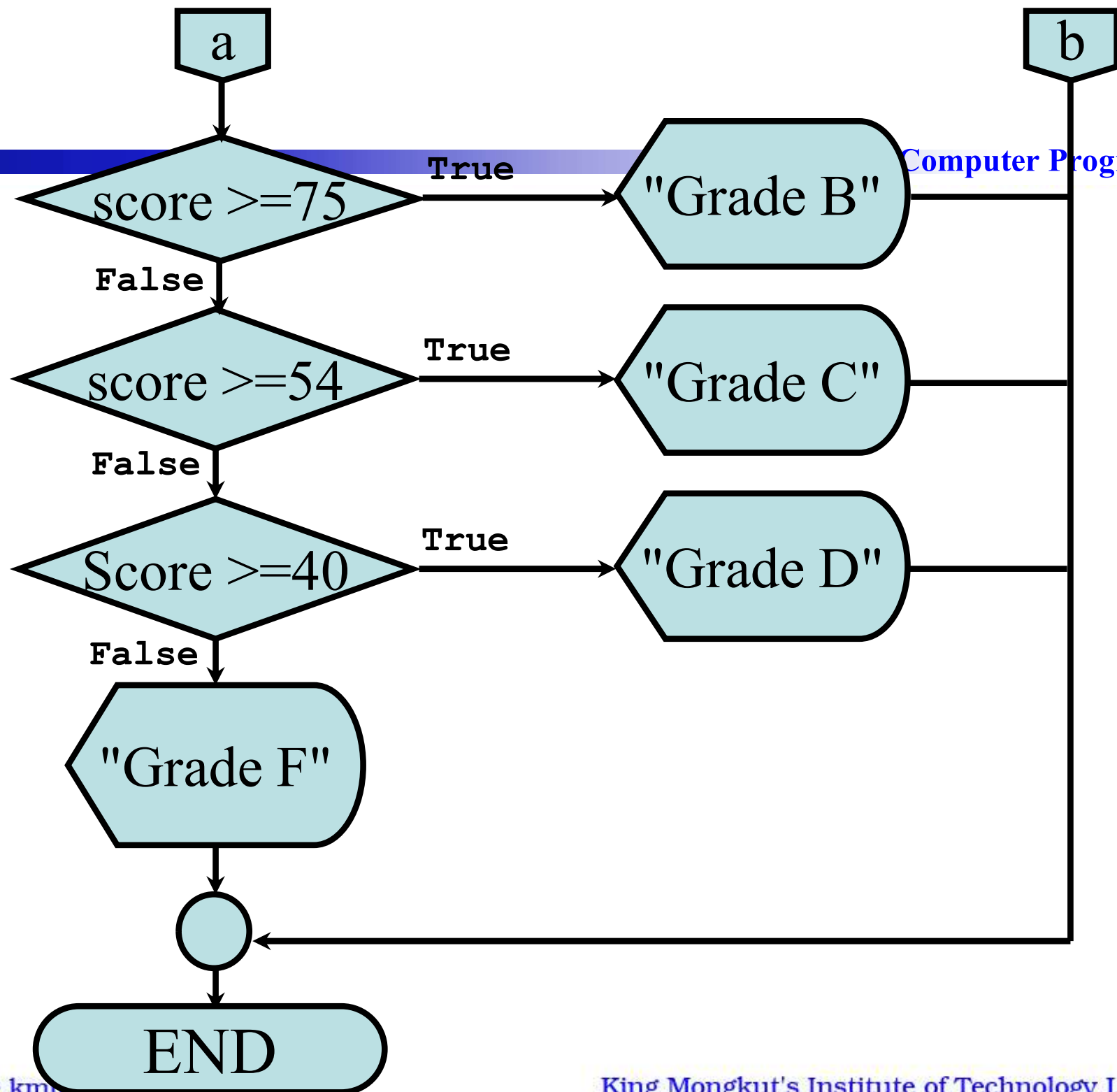
คะแนน 0 – 39.99 ได้เกรด F

คะแนน 68 – 81.99 ได้เกรด B

คะแนน 40 – 53.99 ได้เกรด D

แล้วแสดงผลลัพธ์ ชื่อ คะแนน และเกรด

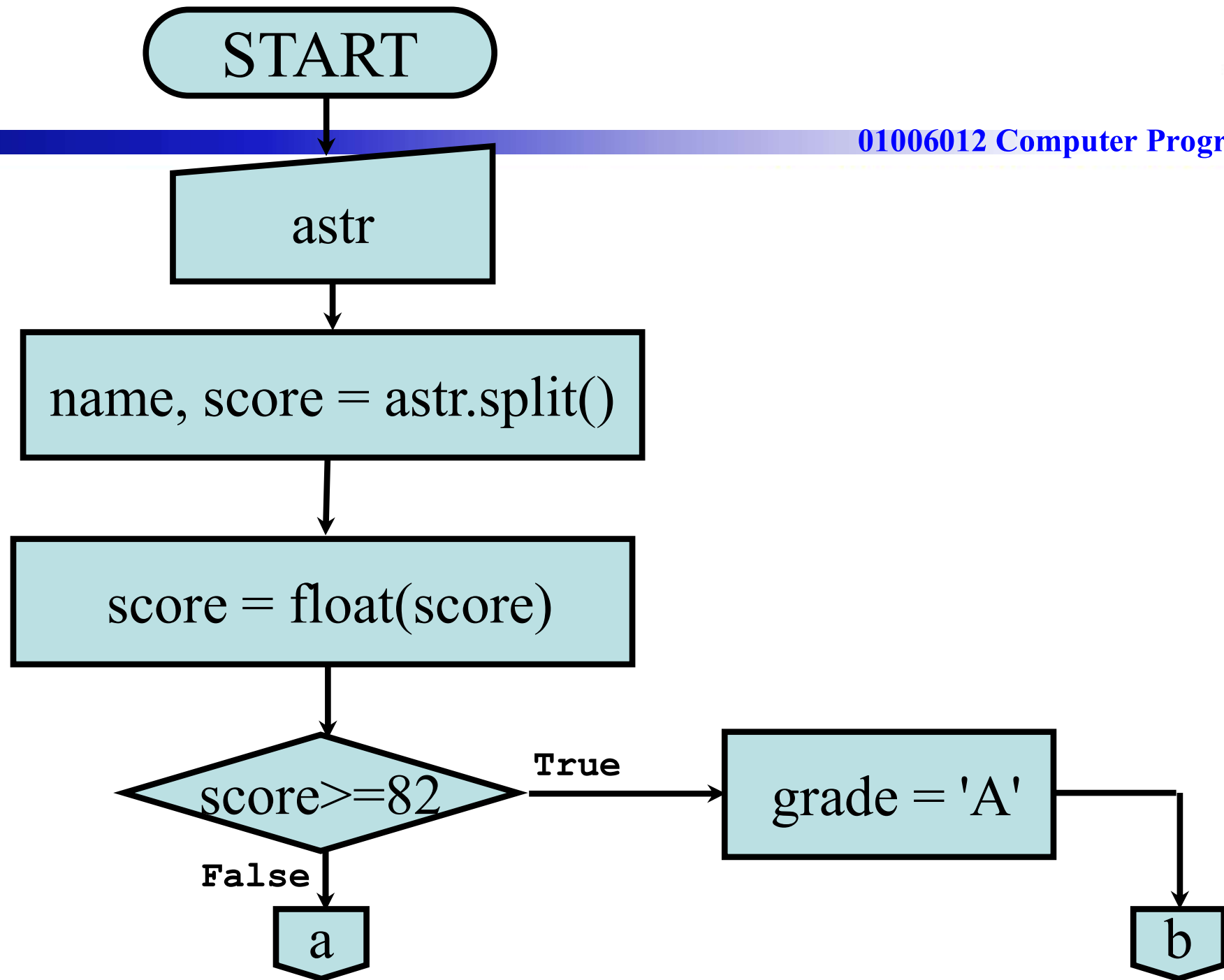


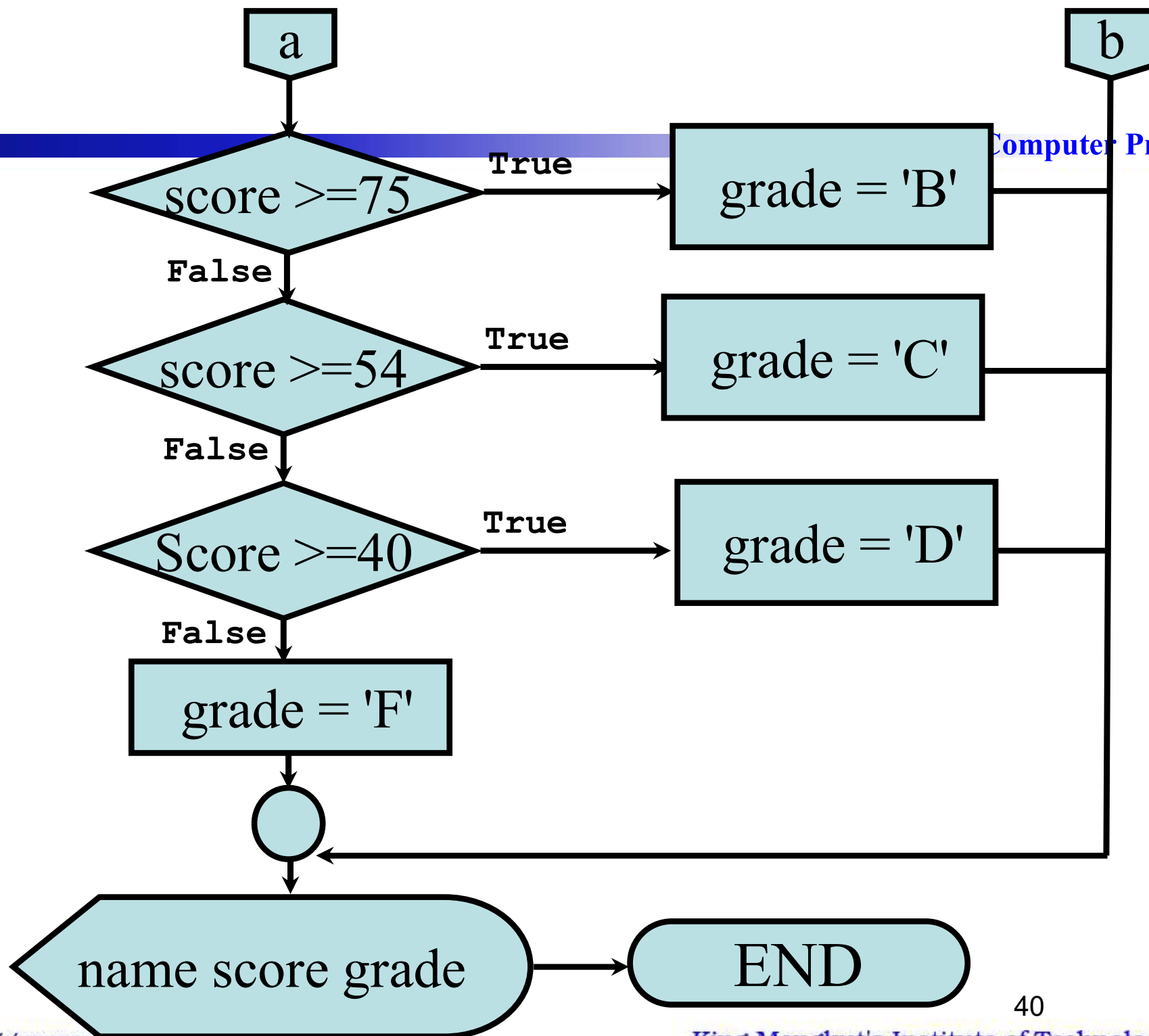


# โปรแกรมตรวจสอบเกรด

01006012 Computer Programming

```
astr = input("Enter name score : ")
name,score = astr.split()
score = float(score)
if score>=82:
    print(f"{name} got {score:.2f} and A grade !!!")
elif score >= 68 :
    print(f"{name} got {score:.2f} and B grade !!!")
elif score >=54:
    print(f"{name} got {score:.2f} and C grade !!!")
elif score >= 40:
    print(f"{name} got {score:.2f} and D grade !!!")
else:
    print(f"{name} got {score:.2f} and F grade !!!")
```







# โปรแกรมตรวจสอบเกรด

01006012 Computer Programming

```
astr = input("Enter name score : ")
name,score = astr.split()
score = float(score)
if score>=82:
    grade = 'A'
elif score >= 68 :
    grade = 'B'
elif score >=54:
    grade = 'C'
elif score >= 40:
    grade = 'D'
else:
    grade = 'F'
print(f"{name} got {score:.2f} and {grade} grade !!!")
```

# โปรแกรมตรวจสอบเกรด

01006012 Computer Programming

Enter name score : Linda 88  
Linda got 88.00 and A grade !!!

Enter name score : Meepoo 75  
Meepoo got 75.00 and B grade !!!

Enter name score : Jim 55.5  
Jim got 55.50 and C grade !!!

Enter name score : Mitree 43  
Mitree got 43.00 and D grade !!!

Enter name score : Aura 39.99  
Aura got 39.99 and F grade !!!

# การเปลี่ยนตัวเลขเป็นข้อความ

01006012 Computer Programming

```
x = 3.1415926  
s = str(x)  
print(s,type(s))
```

3.1415926 <class 'str'>

```
x = 3.1415926  
s = f"{x:0.6f}"  
print(s,type(s))
```

3.141593 <class 'str'>

```
x = 3.1415926*10000  
s = f"{x:,.3f}"  
print(s,type(s))
```

31,415.926 <class 'str'>

# การแสดงผลจำนวนทศนิยม แบบจำนวนเต็ม

01006012 Computer Programming

```
x = 30/2
if int(x) == x :
    print(f"{int(x)} is an integer.")
else:
    print(f"{x} is a float.")
```

15 is an integer.

```
x = 100/32
if int(x) == x :
    print(f"{int(x)} is an integer.")
else:
    print(f"{x} is a float.")
```

3.125 is a float.