Python 2 – Conditionals and loops

Lecture 24 – COMPSCI111/111G SS 2018

Today's lecture

- Recap of yesterday's lecture
- `if` statements
- `while` loops

Recap

- Introduced the IDLE IDE, variables
- Basic arithmetic operators
  - Modulus (%) operator
- `print()` function can be used to display text, arithmetic operations, variables etc.
- `input()` function allows you to capture the user's input from the keyboard
  - `int()` converts the string value from `input()` into an integer
  - `float()` converts the string value from `input()` into a floating point value

Recap

- `int()` and `float()` can also convert integers/floating point numbers to other data types
- Example:

```
x = 20.56
print(int(x))  #output is 20
```

- Example:

```
y = 10
print(float(y))  #output is 10.0
```

IF statement

- Conditional activity (ie. 'if this then do that') is an important part of many programs
- The `if` statement lets you introduce conditional activity into your program
- Statements that are executed when `if` is true must be tabbed underneath the `if` statement
- Syntax:

```
if [logical condition]:
        [lines of code here]
else:
        [lines of code here]
```

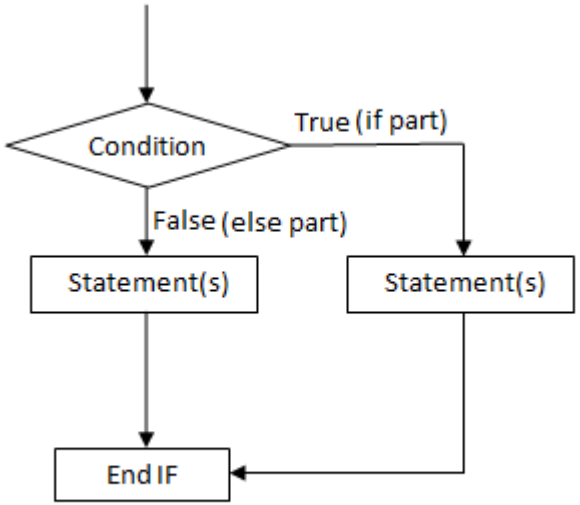
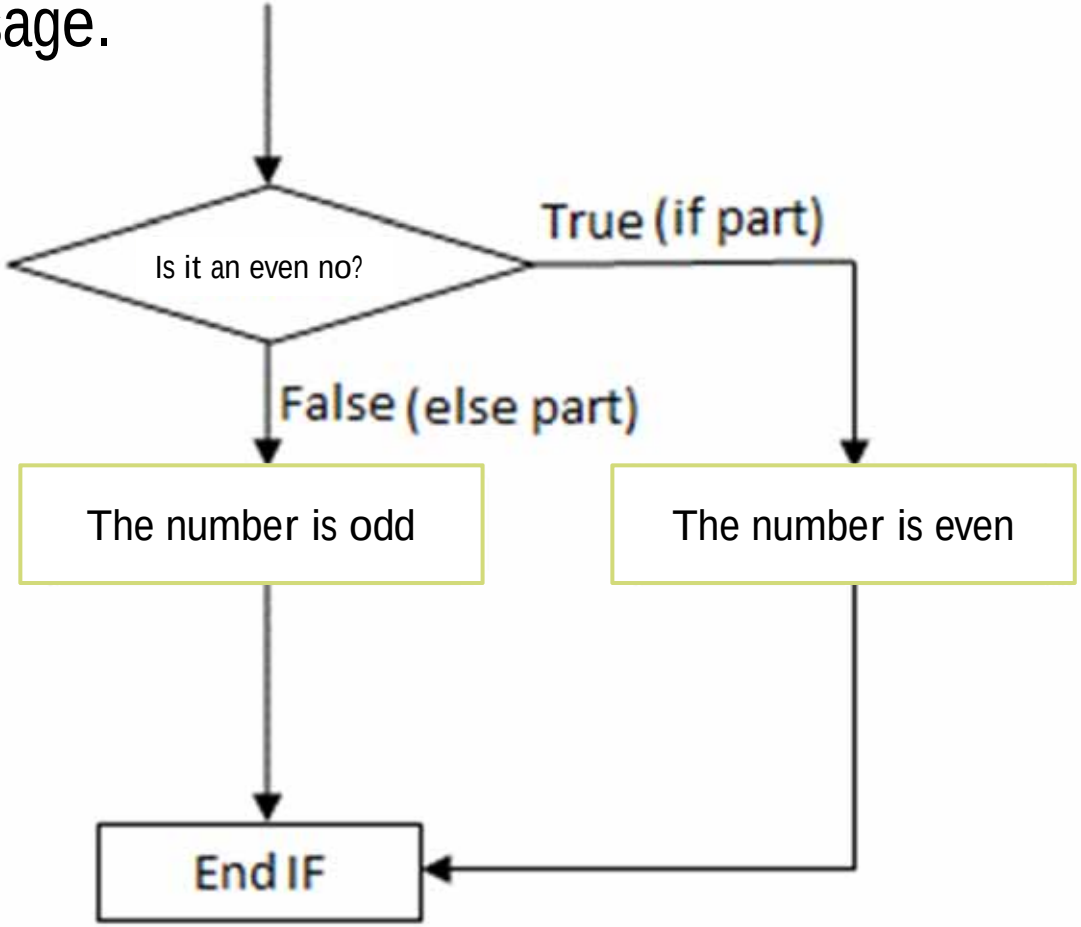


fig: Flowchart for if ... else statement

Example

- determine if a number is odd or even, and print out an appropriate message.



Logical conditions

- A logical condition will either evaluate to true or false

Meaning	Operator	Example
Less than	<	a < b
Less than or equal to	<=	a <= b
Greater than	>	a > b
Greater than or equal to	>=	a >= b
Equal to	==	a == b
Not equal to	!=	a != b

Logical conditions

- You can combine logical conditions using the Boolean operators
- `if a and b:`
  - If the test in a and b evaluate to true, then the logical condition will be true
  - e.g. `if x > 1 and y < 2`
- `if a or b:`
  - If either a or b evaluate to true, then the overall logical condition will be true
  - e.g. `if x > 1 or y < 2`

Logical conditions

- `if not(a):`
  - Inverts the result of a
  - e.g. `if not(5>6)`

IF statement example

L24Demo1.py

- Write a program that asks the user to enter a number between 1 and 10 (inclusive). The program will print out "Correct" if the number is in the range and "Incorrect" if the number is outside the range.
- Example output (bold text is the user's input):

```
Please enter a number (1-10): 34
Incorrect
Please enter a number (1-10): 6
Correct
```

Example:

Prompt the user for a number

Convert it to an integer

If between 1 and 10 (inclusive)

- Print “Correct”

Else

- Print “Incorrect”

IF statement example

L24Demo2.py

```
number = int(input("Please enter a number
(1-10): "))

if number >= 1 and number <= 10:
    print("Correct")
else:
    print("Incorrect")
```

Exercise 1

TRY IT OUT!

L24Ex1.py

- Write a program that asks the user to enter a number. The program should determine if the number is odd or even, and print out an appropriate message.
- Example output (bold text is the user's input) :

```
Please enter a number: 56
You entered 56 which is even
Please enter a number: 33
You entered 33 which is odd
```

Algorithm:

Prompt the user for a number

Convert it to an integer

If it is an even number

- "You entered", _____,"which is even"

Else

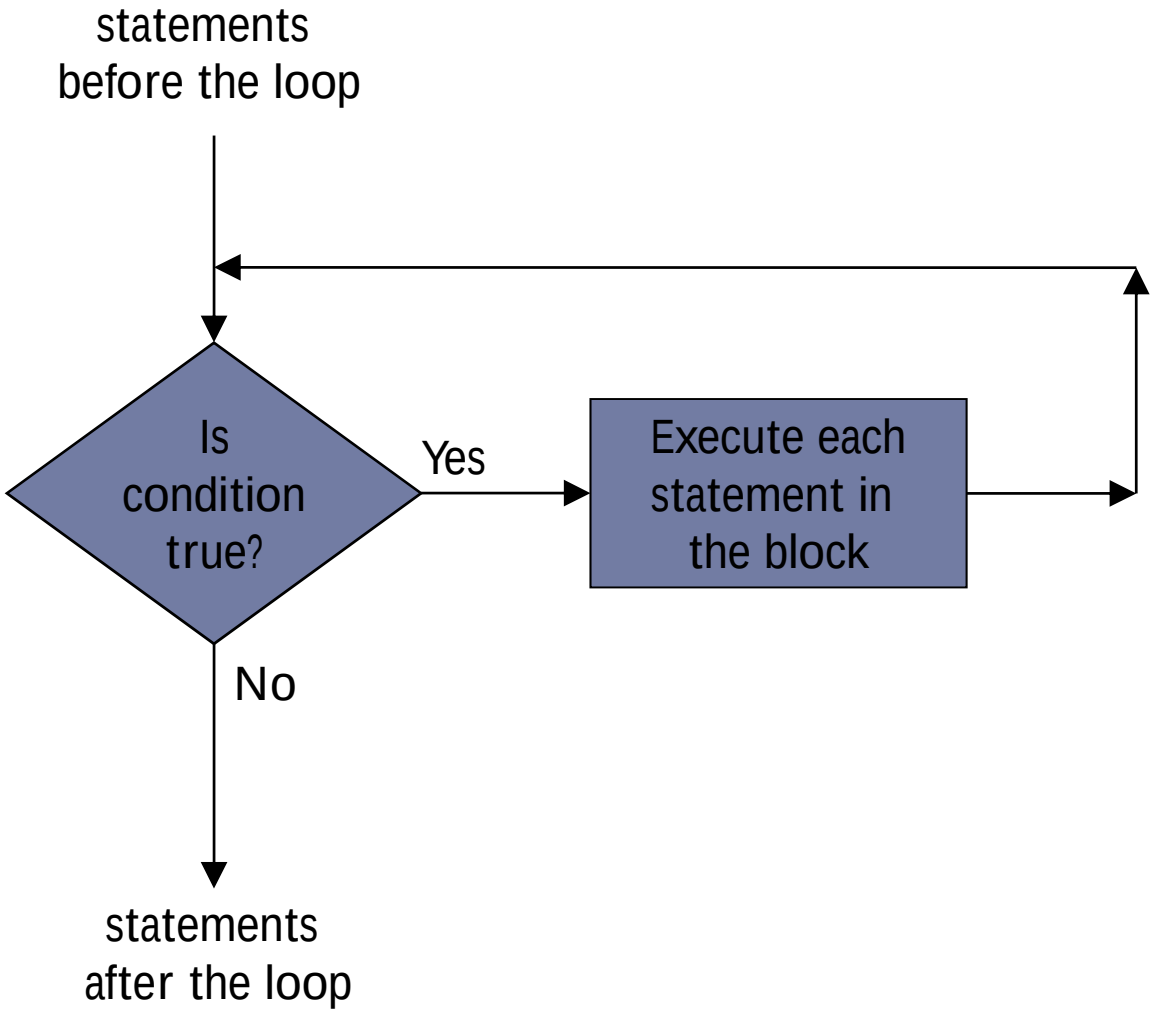
- "You entered", _____,"which is odd"

Loops

- Allows you to repeat certain statements for as long as the loop's logical condition evaluates to true
- Statements that are executed when the `while`'s condition is true must be tabbed underneath the `while` statement
- Syntax:

```
while [logical condition]:
    [lines of code here while condition is true]
```

Loops


statements
before the loop
Is
condition
true?
Yes
Execute each
statement in
the block
No
statements
after the loop

Example

L24Demo3.py

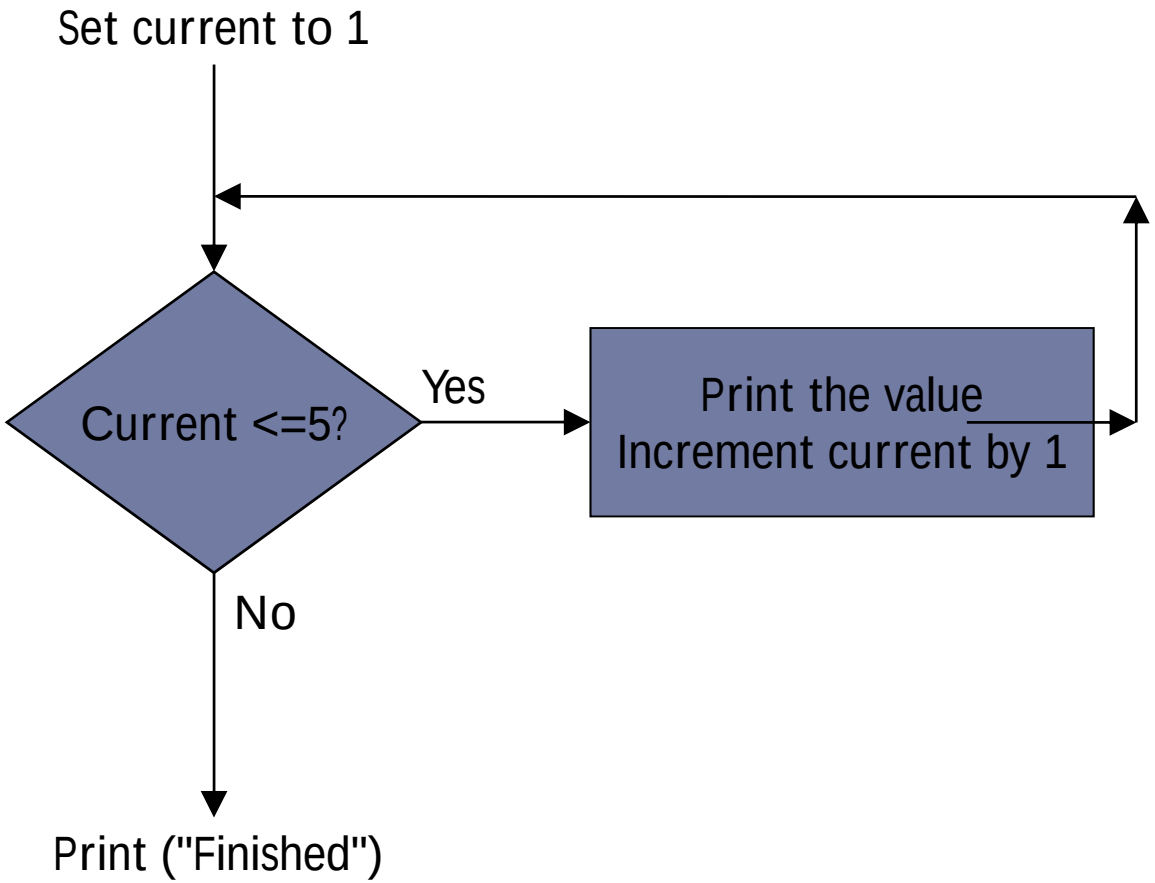
- Write a program to print the numbers from 1 to 5

```
current = 1
while current <= 5:
    print(current)
    current = current + 1
print("Finished!")
```

- Output:

```
1
2
3
4
5
Finished!
```

Loops


Set current to 1
Current <=5?
Yes
Print the value
Increment current by 1
No
Print ("Finished")

Exercise 2

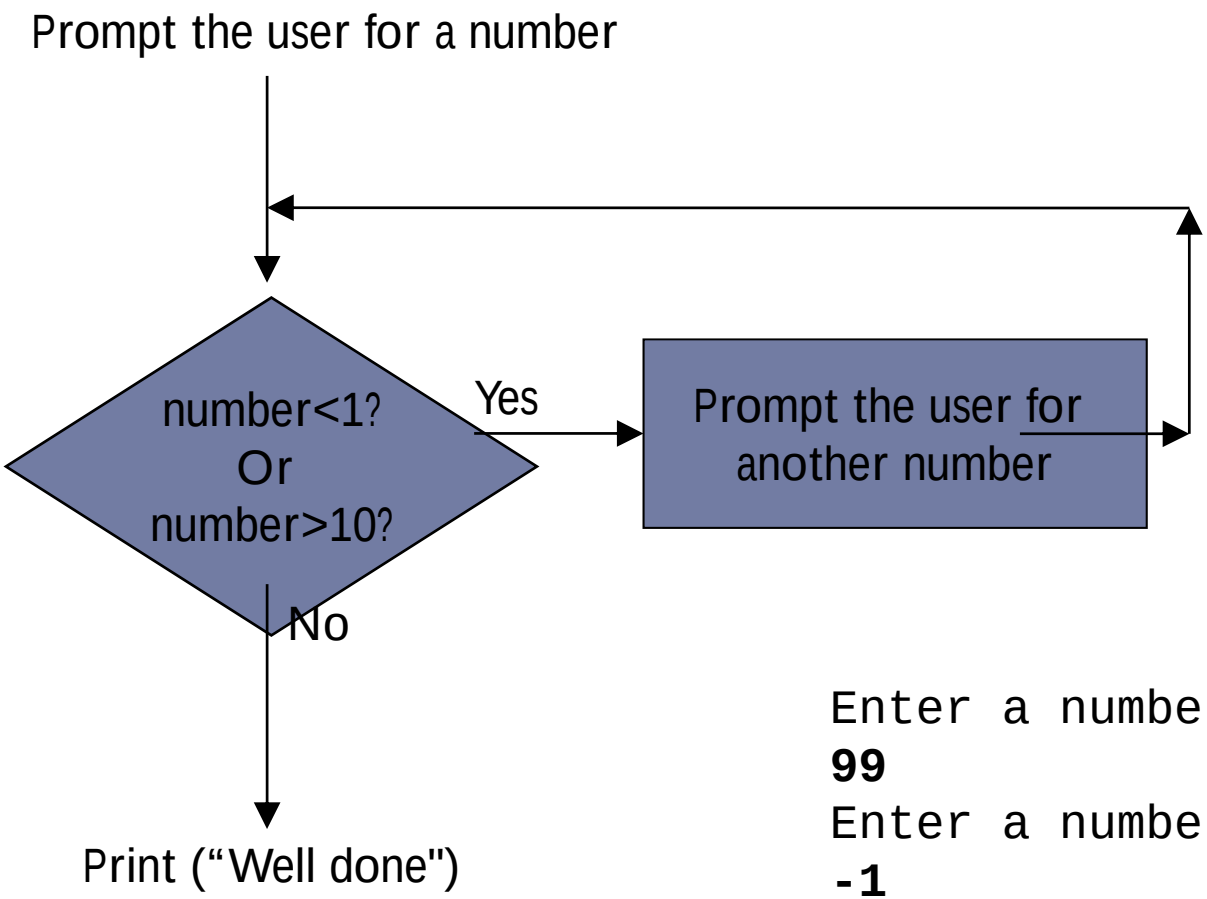
L24Ex2.py

- Write a program that repeatedly asks the user to enter a number between 1 and 10 (inclusive). When they do so, the program should print "Well done" on the screen and end
- Example output (bold text is the user's input):

```
Enter a number between 1 and 10: 99
Enter a number between 1 and 10: -1
Enter a number between 1 and 10: 10
Well done
```

Flow chart

TRY IT OUT!




```
Enter a number between 1 and 10:
99
Enter a number between 1 and 10:
-1
Enter a number between 1 and 10:
10
Well done
```

Exercise 3

L24Ex3.py

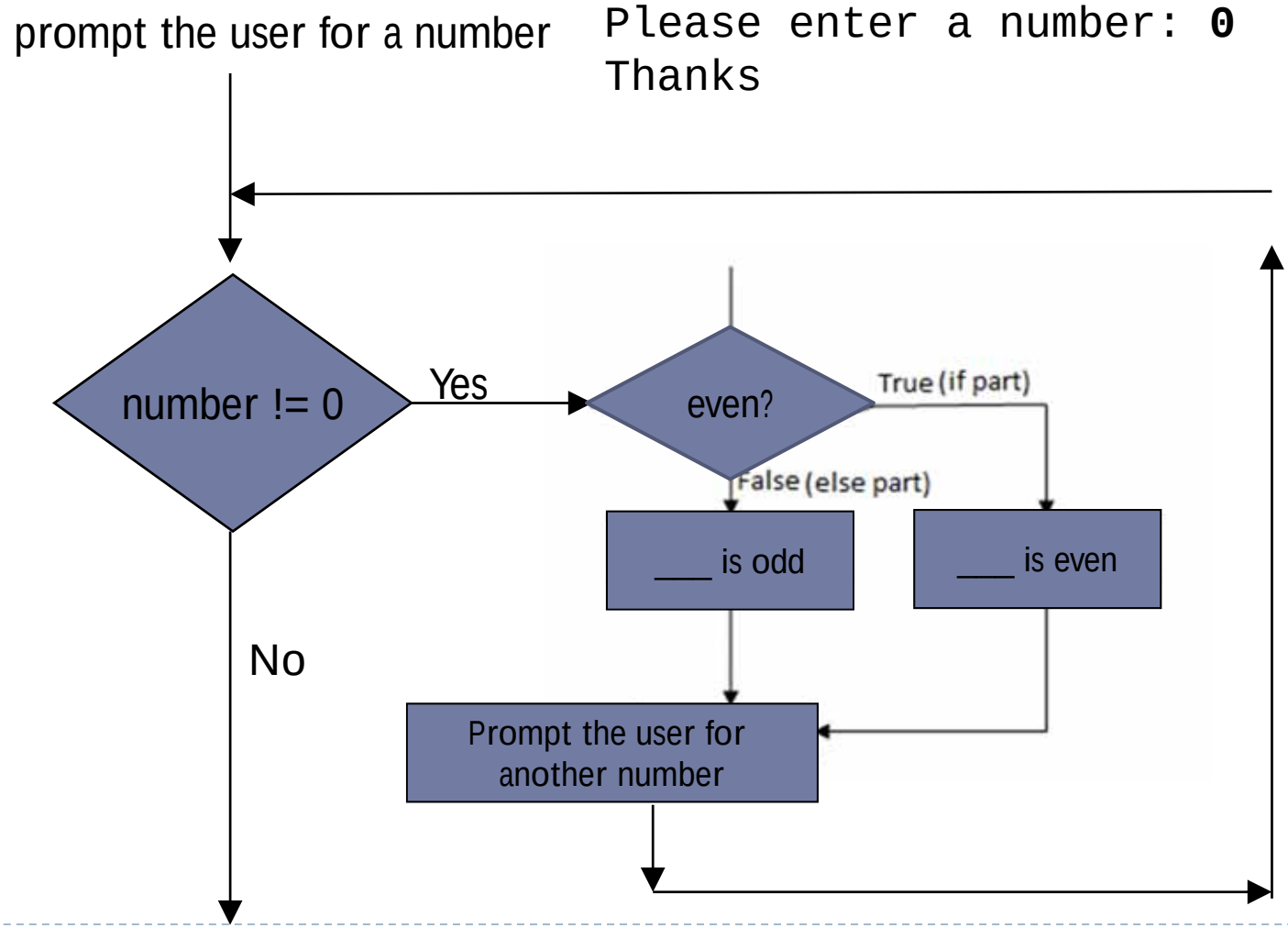
- Write a program that repeatedly asks the user to enter a number. If the number is even, then "x is even" (where x is the number) should be printed on screen. If the number is odd, then "x is odd" should be printed on screen. The program must print "Thanks" and end when the user types '0'
- Example output (bold text is the user's input):

```
Please enter a number: 45
45 is odd
Please enter a number: 12
12 is even
Please enter a number: 0
Thanks
```

Flow chart

```
Please enter a number: 45
45 is odd
Please enter a number: 12
12 is even
Please enter a number: 0
Thanks
```

TRY IT OUT!



Summary

- `if` statements allow you to introduce conditional activities into your program
- `while` loops allow you to repeat certain statements for as long as the logical condition evaluates to true

- Post-Lecture-Quiz: PLQ_24
  - https://coderunner2.auckland.ac.nz/moodle/mod/quiz/view.php?id=630

Exercises

- What is the output of the following statements?

```
temp = 90
if temp >= 90:
        print("hot")
        temp = temp - 10
if temp < 70:
        print("cold")
if temp == 80:
        print("just right")
```