

# Programming Foundations in Python

Adapted From: CMSC 201  
Computer Science I for Majors

Lecture 04 – Decision Structures

# Last Class We Covered

- Python's operators
  - Arithmetic operators
    - Mod and integer division
  - Assignment operators
  - Comparison operators
  - Boolean operators
- The order of operations

## Any Questions from Last Time?

# Today's Objectives

- To get practice using the Boolean data type
- To understand how to use decision structures
  - One-way (using `if`)
  - Two-way (using `if` and `else`)
  - Multi-way (using `if`, `elif`, and `else`)
- To learn about nested decision structures

# Quick Introduction to `main()`

# main()

- In today's class, we introduce the code  
`def main() :`  
as the first line of code in our file
- `main()` is an example of a **function**
- We can use functions to organize our code
  - We'll cover them in detail later

# Functions

- For now, think of functions as something similar to a variable
  - Variables hold data
  - Functions hold code
- We use the variable's name to access its data
- We use the function's name to access its code

# Using `main()` for Your Code

- From now on, use `main()` in your code
  - Every file should have `main()`

```
def main():
```

declaring our `main()` function

```
    className = input("What class is this? ")  
    print(className, "is awesome!")
```


```
main()
```

calling our `main()` function



# Control Structures

# Control Structures

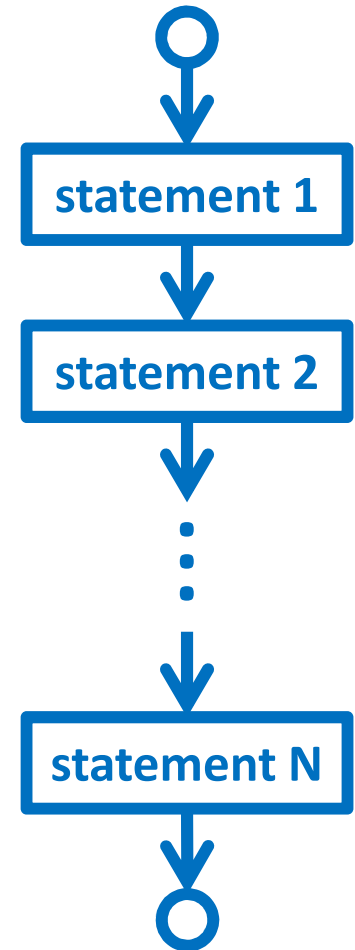
- Structures that control how the program “flows” or operates, and in what order
- Sequence 

we’ve already seen this
- Decision Making

what we’re covering today
- Looping

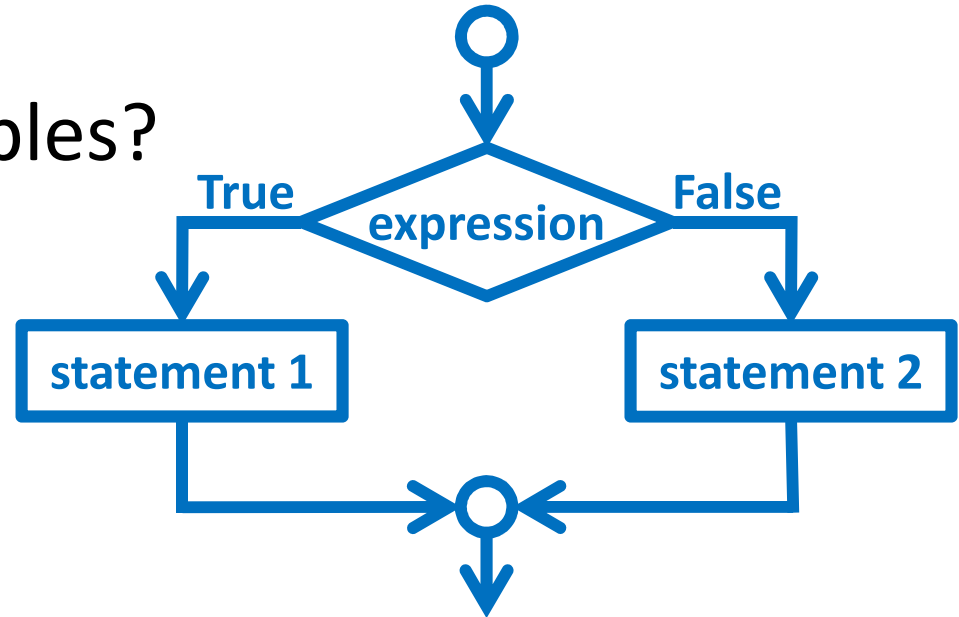
# Sequence

- One step after another,  
with no choices to make
- What are some real life examples?
  - Dialing a phone number
  - Purchasing and paying for groceries



# Decision Making

- Selecting one choice from many based on a specific reason or condition
  - If something is true, do **A** ... if it's not, do **B**
- Some real life examples?
  - What homework to work on tonight
  - Choosing where to eat lunch



# Decision Making

- Why did we spend so much time on comparison operators and logical operators last time?
- Because we can use them to *control* how our program works and what code it runs
  - By using the decision structures

## One-Way Decision Structures

# One-Way Decision Structures

- Decision structures let Python make choices
  - Based on some condition

```
def main():  
    weight = float(input("How heavy is your bag? "))  
    if weight > 50:  
        print("That bag is too heavy.")  
        print("There will be a $25 charge.")  
  
    print("Thank you for your business.")
```

main()

# “if” Statements

- The Python **if** statement is used to implement the decision

```
if <condition>:  
    <body>
```

- The **condition** must evaluate to **True** or **False**
- The **body** is a sequence of one or more statements indented under the **if** heading



# Formatting Decision Structures

- Each **if** statement must close with a colon
  - Two vertical dots ( : )
- Code in the **body** (that is executed as part of the **if** statement) must be indented
  - By four spaces
  - Hitting the “Tab” key in many editors will automatically indent it by four spaces

# “if” Semantics

- The semantics of the **if** should be clear
  - First, the condition in the heading is evaluated
  - If the condition is **True**
    - The statements in the body are executed
    - Control passes to the next statement in the program
  - If the condition is **False**
    - The statements in the body are skipped
    - Control passes to the next statement in the program

# One-Way Decisions

- The body of the **if** either executes or not depending on the condition
- Control then passes to the next (non-body) statement after the **if**
- This is a ***one-way*** or ***simple*** decision



# One-Way Example: Temperature

- You're studying abroad, and need to convert the temperature from Celsius to Fahrenheit

```
def main():  
    celsius = float(input("What is temp in Celsius? "))  
    fahrenheit = 9/5 * celsius + 32  
  
    print("The temperature is", fahrenheit, "degrees Fahrenheit.")  
  
main()
```

# Temperature Warnings

- Let's now modify the program to print a warning when the weather is extreme
- Any temperature that is...
  - Over 95 degrees Fahrenheit
    - Will cause a hot weather warning
  - Lower than 32 degrees Fahrenheit
    - Will cause a cold weather warning

# Temperature Example Code

```
def main():  
    celsius = float(input("What is temp in Celsius? "))  
    fahrenheit = 9 / 5 * celsius + 32  
    print("The temperature is", fahrenheit, "degrees Fahrenheit.")  
    if fahrenheit > 95:  
        print("It's really hot out there, be careful!")  
    if fahrenheit < 32:  
        print("Brrrrrr. Be sure to dress warmly!")  
  
main()
```

# Temperature Example Code

```
def main():  
    celsius = float(input("What is temp in Celsius? "))  
    fahrenheit = 9 / 5 * celsius + 32  
    print("The temperature is", fahrenheit,  
          "degrees Fahrenheit.")  
    if fahrenheit > 95:  
        print("It's really hot out there, be careful!")  
    if fahrenheit < 32:  
        print("Brrrrrr. Be sure to dress warmly!")
```

main()

this is the  
main level of  
our program

this level of the code is  
only executed if  
fahrenheit > 95

this level of the code is  
only executed if  
fahrenheit < 32

## Two-Way Decision Structures



# Two-Way Decisions

- In Python, a ***two-way decision*** can be used by adding an **else** clause onto an **if** clause
- This is called an **if-else** decision structure:

```
if <condition>:  
    <codeA statements>  
else:  
    <codeB statements>
```

# How Python Handles `if-else`

- Python will evaluate the condition, and then...
  - If the condition is **True**, the set of **codeA** statements under the **if** are executed
  - If the condition is **False**, the set of **codeB** statements under the **else** are executed
- The code that comes after the **if-else** is executed only after one of the sets of statements is executed

# Two-Way Code Framework

```
if theCondition:  
    <codeA statements>  
else:  
    <codeB statements>
```



- Only execute **codeA** if **theCondition** is True
- If **theCondition** is not True, run **codeB**

# Simple Two-Way Example

```
def main():  
    x = int(input("What is the value of X? "))  
    if x > 5:  
        print("X is larger than five!")  
    else:  
        print("X is less than or equal to five!")
```

only one of these will execute

main()

this is the  
main level of  
our program

this statement is  
only executed if  
 $x > 5$  is True

this statement is  
only executed if  
 $x > 5$  is False

# Simple Two-Way Example #2

```
def main():  
    num = int(input("Enter a number: "))  
  
    if num % 2 == 0:  
        print("Your number is even.")  
    else:  
        print("Your number is odd.")  
  
    print(num, "is a good number!")  
main()
```

when is this  
line executed?

## Multi-Way Decision Structures

# Bigger (and Better) Decision Structures

- One-way and two-way structures are limited
- What if we have multiple exclusive outcomes?
  - **Exclusive** outcomes do not overlap with each other
  - *e.g.*, value of a playing card, letter grade in a class
- What could we use to represent these?

# “`elif`” Statements

- The Python `elif` statement is used to handle additional exclusive conditions
  - Must have a “starting” `if` statement
  - The `elif` statements must have a *condition*

short for  
“`else if`”



```
if <condition1>:  
    <codeA statements>  
elif <condition2>:  
    <codeB statements>
```



# Multi-Way Code Framework

**if** <condition1>:

    <codeA statements>

**elif** <condition2>:

    <codeB statements>

**elif** <condition3>:

    <codeC statements>

# more "elif" statements if needed

**else:**

    <codeD statements>

as many "elif"  
cases as are needed

"else"  
statement  
is optional



# Multi-Way Decision Example

- A computer science professor gives a five-point quiz at the beginning of every class
- Possible grades are as follows:

5 points: A	3 points: C	1 point: F
4 points: B	2 points: D	0 points: F
- To print out the letter grade based on the raw points, what would the code need to look like?

# Multi-Way Decision Solution

```
def main():  
    score = int(input("Your quiz score out of 5: "))  
    if score == 5:  
        print("You earned an A")  
    elif score == 4:  
        print("You earned a B")  
    elif score == 3:  
        print("You earned a C")  
    elif score == 2:  
        print("You earned a D")  
    else:  
        print("You failed the quiz")
```

```
main()
```

# Multi-Way Decision Solution

```
def main():  
    score = int(input("Your quiz score out of 5: "))  
    if score == 5:  
        print("You earned an A")  
    elif score == 4:  
        print("You earned a B")  
    elif score == 3:  
        print("You earned a C")  
    elif score == 2:  
        print("You earned a D")  
    else:  
        print("You failed the quiz")
```

these are five  
separate statements

since this is an  
if-elif-else  
block, only one of the  
five statements  
will be executed

main()

# Multi-Way Decision Solution

```
def main():  
    score = int(input("Your quiz score out of 5: "))  
    if score == 5:  
        print("You earned an A")  
    elif score == 4:  
        print("You earned a B")  
    elif score == 3:  
        print("You earned a C")  
    elif score == 2:  
        print("You earned a D")  
    else:  
        print("You failed the quiz")
```

How would  
you update  
this to handle  
floats?

main()

# Multi-Way Decision Solution

```
def main():  
    score = float(input("Your quiz score out of 5: "))  
    if score >= 5:  
        print("You earned an A")  
    elif score >= 4:  
        print("You earned a B")  
    elif score >= 3:  
        print("You earned a C")  
    elif score >= 2:  
        print("You earned a D")  
    else:  
        print("You failed the quiz")  
  
main()
```

How would  
you update  
this to handle  
floats?

What would  
happen if we  
just used "if"  
statements  
instead?

# Multi-Way Decision Solution

```
def main():  
    score = float(input("Your quiz score out of 5: "))  
    if score >= 5:  
        print("You earned an A")  
    if score >= 4:  
        print("You earned a B")  
    if score >= 3:  
        print("You earned a C")  
    if score >= 2:  
        print("You earned a D")  
    else:  
        print("You failed the quiz")  
  
main()
```

What would print  
out for a score of 5?

You earned an A  
You earned a B  
You earned a C  
You earned a D

Using only “if” statements  
give us the wrong answer!

# Exclusive Conditions

- Using the **if-elif-else** block lets you have exclusive conditions more easily
  - No need to check `if score < 5 and score >= 4`
- Also, with a block, the **else** is only used if none of the other conditionals are **True**
- The **if-elif-else** block is evaluated from the top down, so the order of statements does matter



## Nested Decision Structures

# Nested Decision Structures

- Up until now, we have only used a single level of decision making
- What if we want to make decisions within decisions?
- These are called ***nested*** decision structures

# Nested Decision Structures

- Python allows you to nest decision structures
  - As many levels deep as you want
  - Nesting can occur inside **if**, **elif**, or **else** statements
- Only “rule” is that every inside level must start with an “**if**”
  - Having matching **elifs** or an **else** is not required



# Nested Decision Structure Example

- For example, we may
  - Ask the user if they have a pet
  - **if** they have a pet
    - Ask the user what type of pet
    - **if** they have a dog, take it for a walk
    - **elif** they have a cat, clean the litter box
    - **else** clean the cage/stable/tank

# Nested Decision Structure Example

```
def main():  
    ans = input("Do you have a pet? (yes/no) ")  
  
    if ans != "no":  
        pet = input("What kind of pet do you have? ")  
  
        if pet == "dog":  
            print("Take it for a walk")  
        elif pet == "cat":  
            print("Clean the litter box")  
        else:  
            print("Clean the cage/stable/tank")  
  
main()
```

# Nested Decision Structures Code

```
if <condition1>:  
    if <condition2>:  
        <codeA statements>  
    elif <condition3>:  
        <codeB statements>  
    else:  
        <codeC statements>  
else:  
    <codeD>
```

# Nested Decision Structures Code

```
if <condition1>:  
    if <condition2>:  
        <codeA statements>  
    elif <condition3>:  
        <codeB statements>  
    else:  
        <codeC statements>  
else:  
    <codeD>
```

this is the main level  
of our program:  
an if-else block

this is the next level,  
inside the first  
if statement

codeA, codeB, and codeC  
are separate statements

since this is an  
if-elif-else  
block, only one of them  
will be executed

if condition1 was false,  
Python will go straight here  
and execute codeD

# Nested Decision Structure Example

- You recently took a part-time job to help pay for your student loans at a local cell phone store
- If you sell at least \$1000 worth of phones in a pay period, you get a bonus
  - Your bonus is 3% if you sold at least 3 iPhones, otherwise your bonus is only 2%



# Nested Decision Solution

```
def main():
    totalSales = float(input("Please enter your total sales:"))

    if totalSales >= 1000.00:
        # only ask this if they are eligible for a bonus
        iPhonesSold = int(input("Enter the number of iPhones sold:"))

        if iPhonesSold >= 3:
            bonus = totalSales * 0.03
        else:
            bonus = totalSales * 0.02

        print("Your bonus is $", bonus)

    else:
        print("Sorry, you do not get a bonus this pay period.")

main()
```

# Image Sources

- One way sign (adapted from):
  - <https://pixabay.com/p-438122>
- Splitting arrow:
  - <https://pixabay.com/p-154512/>
- Three decisions:
  - <https://pixabay.com/p-1020289/>
- Nest with eggs (adapted from):
  - <https://pixabay.com/p-1485378>