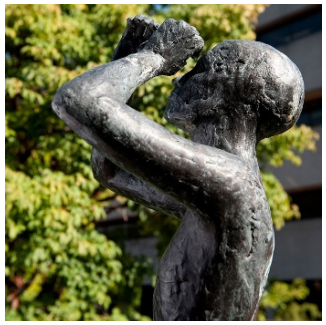


CS1117 – Introduction to Programming

Dr. Jason Quinlan,

School of Computer Science and Information Technology

A TRADITION OF INDEPENDENT THINKING

UCC

University College Cork, Ireland

Coláiste na hOllscoile Corcaigh

Announcements

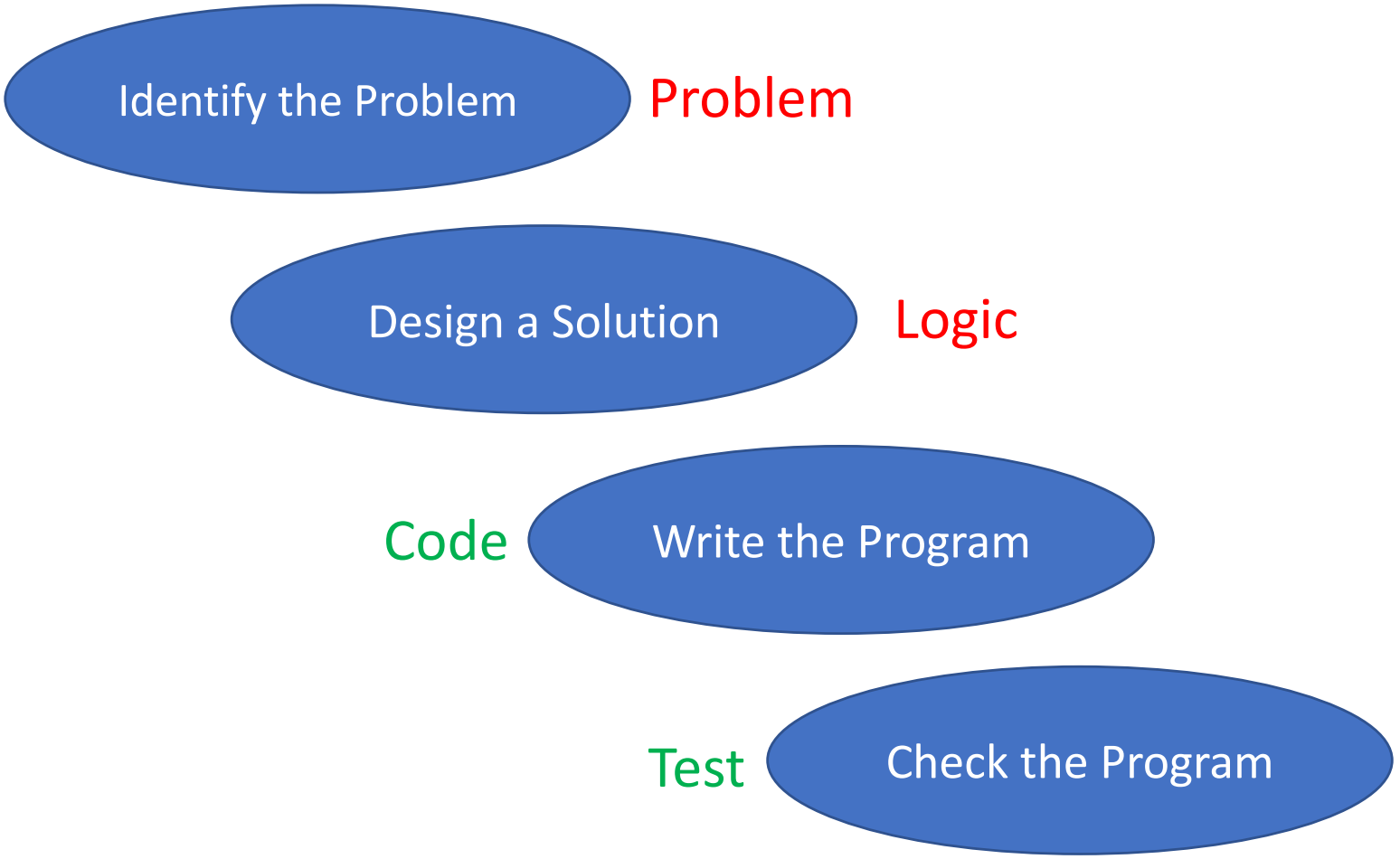
BSc DSA students

If you missed induction

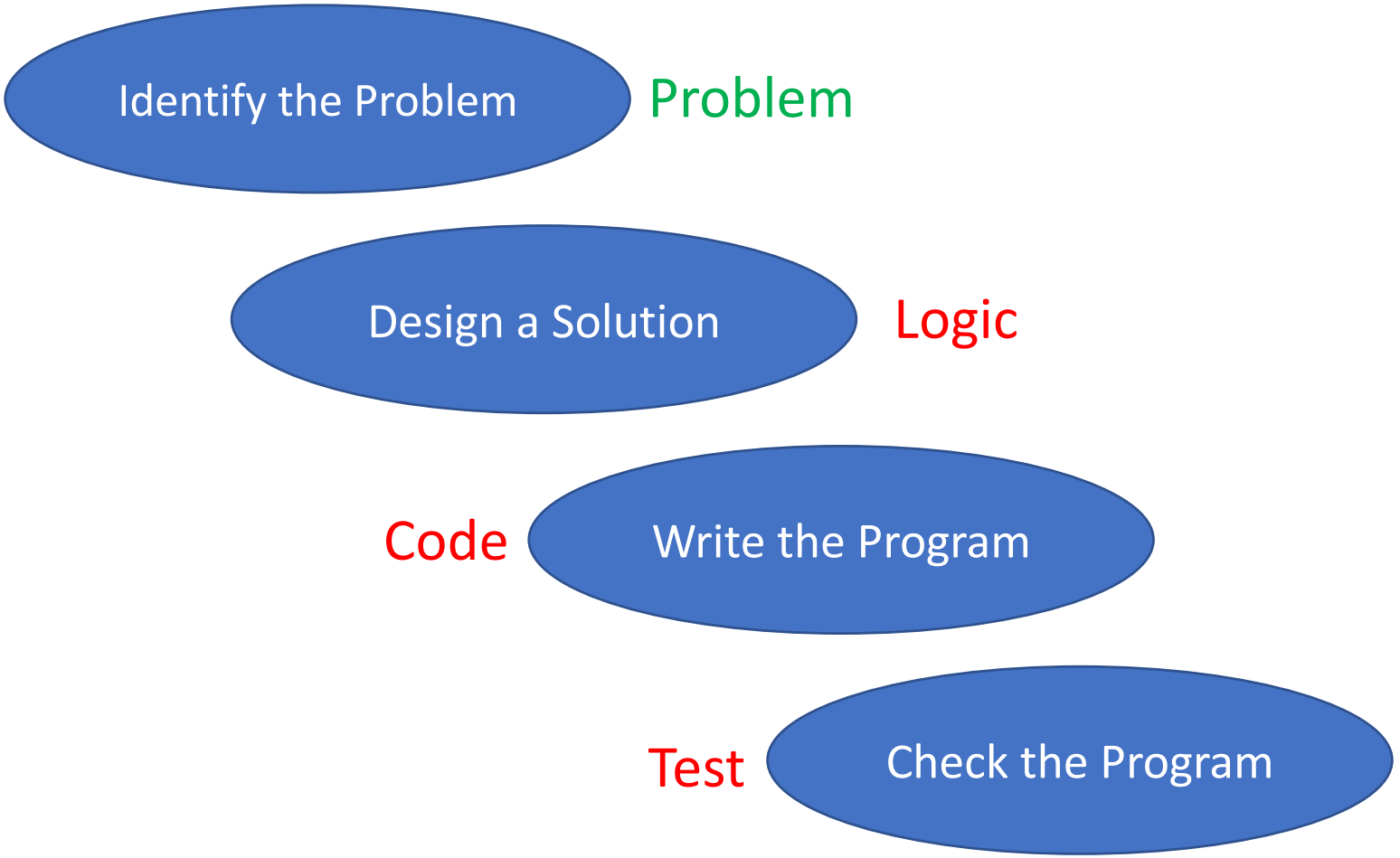
Collect information sheets

Before you leave class

Recap:4 Pillars of Programming



4 Pillars of Programming



Simple Question

I want to know the average age of my class, and I don't want to use JAVA :)

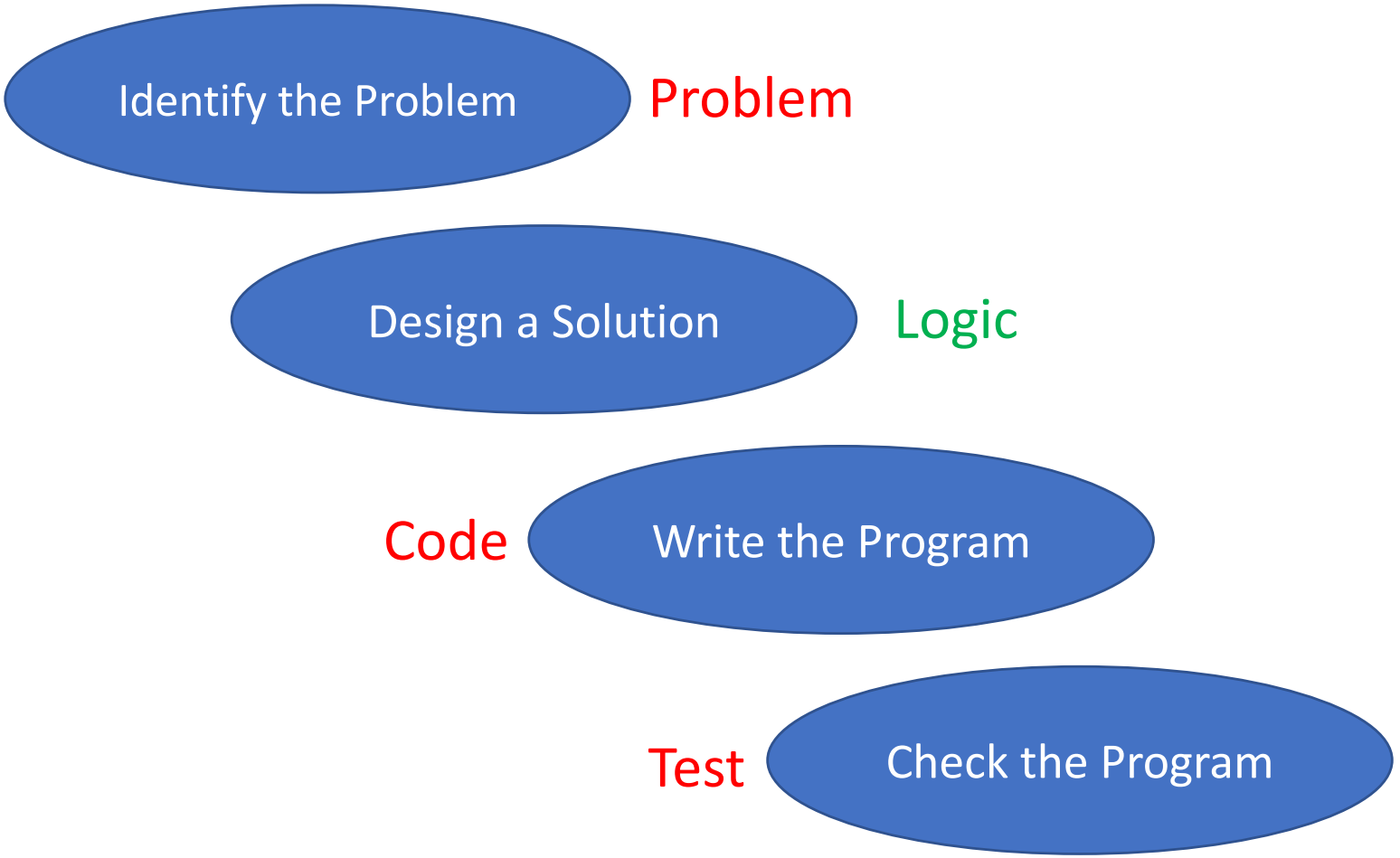
Simplify this, let's start with two people.

In my program, I want:

- To be able to enter the ages of two people.
- Have the computer calculate their average age.
- Display the answer.

Before I write the code, I want to make sure the logic is correct. So, I need to design a solution.

4 Pillars of Programming



Pseudocode

Pseudocode is an informal high-level description of a computer program or algorithm.

It is written in symbolic text which must be translated into a programming language before it can be executed.

Pseudocode makes its easy to write out the logic without worrying about coding

So, take our average age problem and let's write out the logic for how we would solve it

Simple Question

I want to know the average age of my class. Let's start with two people.

In my program, I want:

- To be able to enter the ages of two people.
- Have the computer calculate their average age.
- Display the answer.

Before I write the code, I want to make sure the logic is correct.

Pseudocode

Input

- display a message asking the user to enter the first age
  - save the first age taken from the keyboard
- display a message asking the user to enter the second age
  - save the second age taken from the keyboard

Processing

- calculate the answer by adding the two ages together and dividing by two

Output

- display the answer on the screen pause so the user can see the answer

Pseudocode

Input

- display a message asking the user to enter the first age
  - save the first age from the keyboard
- display a message asking the user to enter the second age
  - save the second age from the keyboard

Processing

- calculate the average by adding the two ages together and dividing by two

Output

- display the answer on the screen

What our Code looks like in Python

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

```
Jasons-MacBook-Pro:code_snippets jasonquinlan$ python3 ./average_ages.py
Please enter age 1: 4
Please enter age 2: 8
The average age is 6
Jasons-MacBook-Pro:code_snippets jasonquinlan$
```

Run our code from the command line (Terminal)

Software & Resources

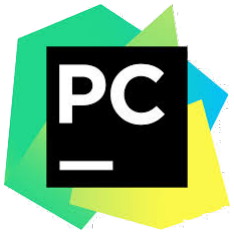
Python Versions:

Python 2 – current version 2.7.15+

Python 3 – current version 3.6.8

We will use Python 3

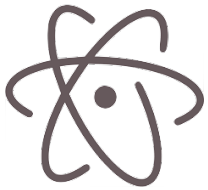
Python IDE:



JetBrains PyCharm:

https://www.jetbrains.com/pycharm/

Atom:



https://atom.io

Naming a file

File is called “average_ages.py”

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

Finicky

Syntax and structure matters! Every dot, comma and space is important. Beware of typos!

***.py* File Extension**

All python programs have a *.py* file extension. Make sure you do not give it a double file extension by mistake. For example *.py.txt*

Variables and Functions

Variables and Functions

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

Realistically all coding can be defined as a mixture of **variables** and **functions**

- age1 is an example of a variable
- print() is an example of a function

Variables and Functions

Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

It can be more useful to store data in a **variable**

- a small piece of allocated memory tagged with a sensible name or label

age1 is a variable we use to save the age input from the keyboard (user)

Variables and Functions

Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

Later in the program:

you can refer to this variable by it's name rather than by its value

average = (age1+age2)/2

Notice BOMDAS - (age1+age2)/2

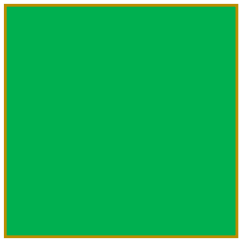
Variables and Functions

No Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

Variables and Functions



No Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

```
print("The average age is %d" % ((int(input("Please enter age 1: "))
                                  + int(input("Please enter age 2: ")))/2))
```

Variables and Functions

No Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

```
print("The average age is %d" % ((int(input("Please enter age 1: "))
                                  + int(input("Please enter age 2: ")))/2))
```

Variables and Functions

Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

As Software Developers , variables make our life easier

Makes the code more manageable

Easier to comment

Easier to understand

And so much easier to debug (the process of finding and removing BUGS...)

Variables and Functions

Variables

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

As stated, syntax and structure matters:

So age1 is not the same as:

Age1 or AGE1 or age_1 or ageOne or any combination of these…

Variables and Functions

Functions

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

input(), int(), and print() are examples of functions:

Method name

String literal in quotes (called an argument)

opening and closing parentheses ( )

input("Please enter age 1: ”)

Variables and Functions

Functions

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

input() and int() are examples of functions that return a value
While print() is an examples of a function that does not

int(input("Please enter age 1: ”)) is an example of a nested function call

Variables and Functions

Nested Functions

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

age1 = int(input("Please enter age 1: ”))

Can be rewritten as

input_age = input("Please enter age 1: ”)
age1 = int(input_age)

Variables and Functions

Nested Functions

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

age1 = int(input("Please enter age 1: ”))

Can be rewritten as

input_age = input("Please enter age 1: ”)
age1 = int(input_age)

Placing a function per line provides clarity to your program, making easier for other coders

Comments in Python

Python ignores everything after the # symbol

average_ages.py

```
# get the first age
age1 = int(input("Please enter age 1: "))
# get the second age
age2 = int(input("Please enter age 2: "))
# determine the average age
average = (age1+age2)/2
# print to screen
print("The average age is %d" % average)
```

Programmers comment their code so others can 'read' and understand it.

is used for a single line comment
''' to open and ''' to close a paragraph of comments.

Insert comments wherever makes sense (except mid Python command).

Comments in Python

Python ignores everything after the # symbol

average_ages.py

```
# get the first age
age1 = int(input("Please enter age 1: "))
# get the second age
age2 = int(input("Please enter age 2: "))
# determine the average age
average = (age1+age2)/2
# print to screen
print("The average age is %d" % average)
```

Programmers comment their code so others can 'read' and understand it.

is used for a single line comment
''' to open and ''' to close a paragraph of comments.

```
'''
get the first age
'''
age1 = int(input("Please enter age 1: "))
```

Insert comments wherever makes sense (except mid Python command).

Keyboard input

For variable assignment everything in code runs right to left

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

- We look at input("Please enter age 1: ”) first
- Then int()
- Then =
- Then we save to the variable age1

- Similar to BOMDAS
  - Operation enclosed in the inner most parentheses are performed first

Keyboard input

Input(string) is used to get input from the keyboard

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

- input() is a function provided by Python to get input from the keyboard
- input is the function name (also known as a method)
- The ( ) provide an area in the code in which you can pass arguments
- "Please enter age 1: " is known as a string literal
- The input from the keyboard is always returned as a "string"
- In our example we entered 4 on the keyboard and input() returns "4"

Keyboard input

Input(string) is used to get input from the keyboard

```
age1 = input("Please enter age 1: ")
# get the second age
age2 = input("Please enter age 2: ")
# determine the average age
average = (age1+age2)/2
# print to screen
print("The average age is %d" % average)
```

- Removing the int() function for now:
- As age1 is just a variable and a variable can have any value
- We can save the value of age1 as “4”
- But once we get to the average variable assignment
- We are trying to divide a string by the integer 2
- error

Keyboard input

Input(string) is used to get input from the keyboard

```
age1 = input("Please enter age 1: ")
# get the second age
age2 = input("Please enter age 2: ")
# determine the average age
average = (age1+age2)/2
# print to screen
print("The average age is %d" % average)
```

```
Traceback (most recent call last):
  File "./average_ages.py", line 9, in <module>
    average = (age1+age2)/2
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Casting

We want to save our input as an integer

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

- int() is a function provided by Python to cast a value to an integer
- int is the function name
- The ( ) provide an area in the code in which you can pass arguments
- In this instance we pass the returned value of

  input("Please enter age 1: ") = "4"
- The return value from int() is always returned as an integer unless

Casting

We want to save our input as an integer

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

The value being cast can not be converted to an integer, examples:

- “a” – any alphabetical value

```
input_age = int("a")
print(input_age)
```

Casting

We want to save our input as an integer

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

The value being cast can not be converted to an integer, examples:

- "a" – any alphabetical value

```
input_age = int("a")
print(input_age)
```

Semantic (logic) error

```
Traceback (most recent call last):
  File "./average_ages.py", line 12, in <module>
    input_age = int("a")
ValueError: invalid literal for int() with base 10: 'a'
```

Mathematical expression

Calculate the average age

average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

- This line is self explanatory
- We take the two input ages
- Divide by 2
- And save the calculated value in the average variable

Print

Display the answer on the screen

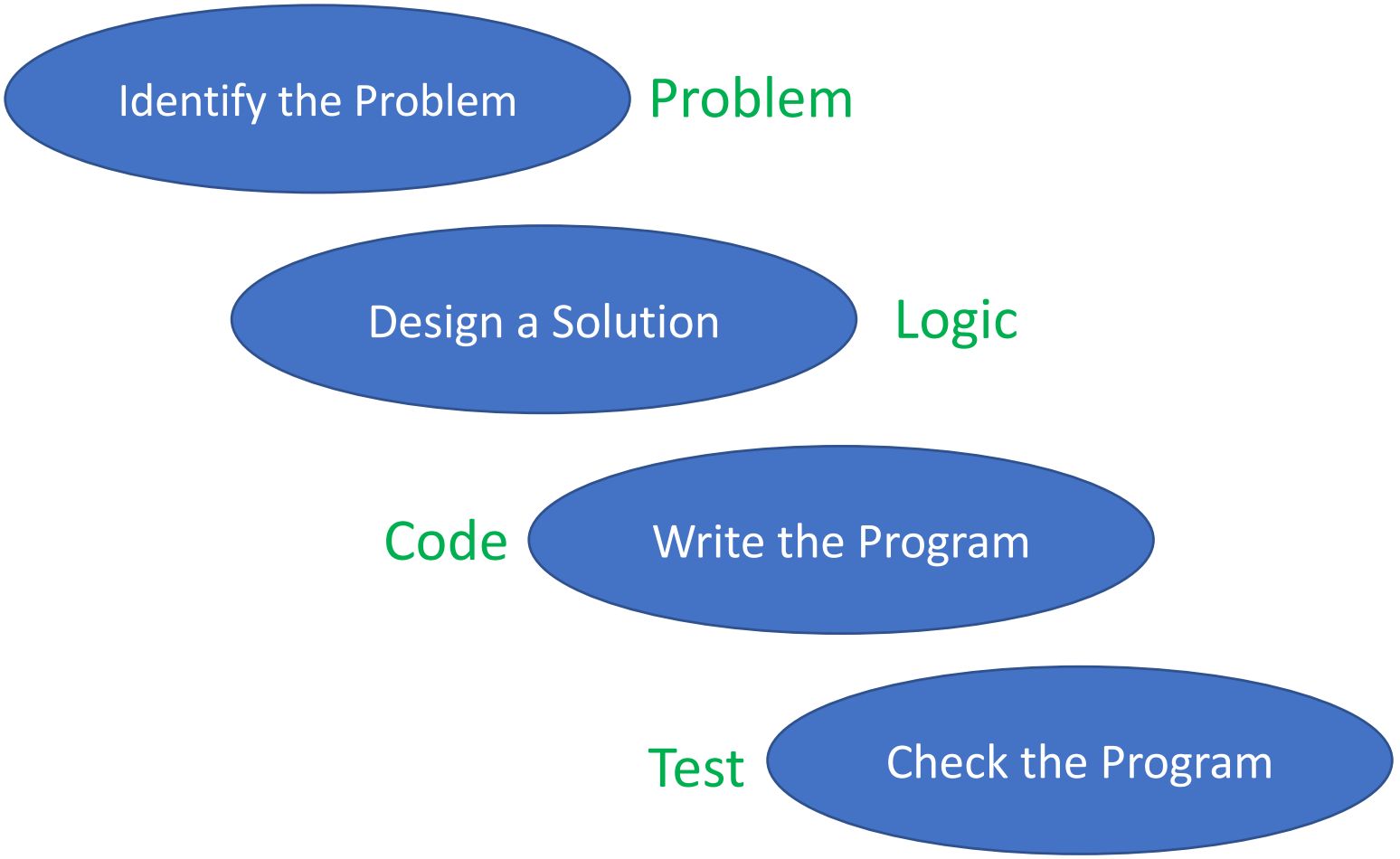
average_ages.py

```
1  # get the first age
2  age1 = int(input("Please enter age 1: "))
3  # get the second age
4  age2 = int(input("Please enter age 2: "))
5  # determine the average age
6  average = (age1+age2)/2
7  # print to screen
8  print("The average age is %d" % average)
```

- print() allows us to output content, typically to the screen
- The %d" % average is a mechanism we can use to format content

- For now, print() does exactly what it says

Recap:4 Pillars of Programming



Announcements

Do some coding in Atom
at the end of class

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh