

Program State and Program Execution

CSE 1310 – Introduction to Computers and Programming

Understanding a Line of Code

- To understand a line of the code, you must be able to answer the following three questions:
 - which variables does this line affect, and how?
 - does it create any new variables?
 - does it change the value of any existing variables?
 - what line of code will be executed NEXT?
 - we have not seen examples yet, but in many cases the next line to be executed will NOT be simply be the next line in the text file.
 - what **side effects** does this line cause?
 - printing to the screen
 - (later) creating and modifying files
 - (in future courses) playing audio/video, sending data to a network...

Determining the Effects of a Line

- What will this line do?

```
celsius = (fahrenheit - 32) * 5/9
```

Determining the Effects of a Line

- What will this line do?

```
celsius = (fahrenheit - 32) * 5/9
```

- Clearly, it will change the value of the celsius variable.
- But, what will be the new value?

Determining the Effects of a Line

- What will this line do?

```
celsius = (fahrenheit - 32) * 5/9
```

- Clearly, it will change the value of the celsius variable.
- But, what will be the new value?
- We do not have enough information to determine that.
- To know EXACTLY the effects of a line, we need to know the **program state**.

Defining a Program State

- To figure out what a line of code does, we need to be able to answer these questions:
 - how the line will affect variables.
 - what the next line will be.
 - what side effects will be produced.
- At this early stage in the course, the state of the program consists of two things:
 - which line are we executing right now?
 - what variables exist right now, and what are the values of those variables?

Example of Tracing Program Execution

Line 1: `fahrenheit_string = input("Enter the temperature in Fahrenheit: ")`

Line 2: `fahrenheit = float(fahrenheit_string)`

Line 3: `celsius = (fahrenheit - 32) * 5/9`

Line 4: `print("The temperature in celsius is", celsius)`

CURRENT PROGRAM STATE

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")
Line 2: fahrenheit = float(fahrenheit_string)
Line 3: celsius = (fahrenheit - 32) * 5/9
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 1

TABLE OF VARIABLES

What does this line do?

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")
Line 2: fahrenheit = float(fahrenheit_string)
Line 3: celsius = (fahrenheit - 32) * 5/9
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 1

TABLE OF VARIABLES

What does this line do?

Since this line receives user input, we need to make up some value for this input. So, let's assume that the user enters "59".

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")
Line 2: fahrenheit = float(fahrenheit_string)
Line 3: celsius = (fahrenheit - 32) * 5/9
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 1

TABLE OF VARIABLES

SIDE EFFECT:

prints "Enter the temperature in Fahrenheit:"

NEXT PROGRAM STATE

current line: 2

TABLE OF VARIABLES

fahrenheit_string = '59'

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")
Line 2: fahrenheit = float(fahrenheit_string)
Line 3: celsius = (fahrenheit - 32) * 5/9
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 2

TABLE OF VARIABLES

fahrenheit_string = '59'

What does this line do?

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")  
Line 2: fahrenheit = float(fahrenheit_string)  
Line 3: celsius = (fahrenheit - 32) * 5/9  
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 2

TABLE OF VARIABLES

fahrenheit_string = '59'

NEXT PROGRAM STATE

current line: 3

TABLE OF VARIABLES

fahrenheit_string = '59'
fahrenheit = 59

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")  
Line 2: fahrenheit = float(fahrenheit_string)  
Line 3: celsius = (fahrenheit - 32) * 5/9  
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 3

TABLE OF VARIABLES

fahrenheit_string = '59'
fahrenheit = 59

What does this line do?

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")  
Line 2: fahrenheit = float(fahrenheit_string)  
Line 3: celsius = (fahrenheit - 32) * 5/9  
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 3

TABLE OF VARIABLES

fahrenheit_string = '59'
fahrenheit = 59

NEXT PROGRAM STATE

current line: 4

TABLE OF VARIABLES

fahrenheit_string = '59'
fahrenheit = 59
celsius = 15

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")
Line 2: fahrenheit = float(fahrenheit_string)
Line 3: celsius = (fahrenheit - 32) * 5/9
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 4

TABLE OF VARIABLES

fahrenheit_string = '59'
fahrenheit = 59
celsius = 15

What does this line do?

Example of Tracing Program Execution

```
Line 1: fahrenheit_string = input("Enter the temperature in Fahrenheit: ")  
Line 2: fahrenheit = float(fahrenheit_string)  
Line 3: celsius = (fahrenheit - 32) * 5/9  
Line 4: print("The temperature in celsius is", celsius)
```

CURRENT PROGRAM STATE

current line: 4

TABLE OF VARIABLES

```
fahrenheit_string = '59'  
fahrenheit = 59  
celsius = 15
```

SIDE EFFECT:

prints "The temperature in celsius is 15.0"

NEXT PROGRAM STATE

current line: end of program

TABLE OF VARIABLES

```
fahrenheit_string = '59'  
fahrenheit = 59  
celsius = 15
```

Understanding a Line of Code

- Understanding a line of code means that, given the current program state, you can describe:
 - how the line will change the program state.
 - what side effects the line will produce.

Understanding a Program

- Understanding a program means that you understand how every single line of the program affects the program state.
- Conversely, if you do not understand precisely how a specific line changes program state, **you do not understand the program.**

Understanding a Program

- Understanding a program means that you understand how every single line of the program affects the program state.
- Conversely, if you do not understand precisely how a specific line changes program state, **you do not understand the program.**
- **If this ever happens, it is a sign that you are falling behind, and need to catch up.**