

Comp 150 - Introduction to Computing Programming Basics

Dr. William L. Honig
Associate Professor
Department of Computer Science

Question TWO

What is a “Computer Program”?

Or, what does a programmer do?
Or, how do I write a program?

The Software Development Process

- The process of creating a program is often broken down into stages according to the information that is produced in each phase.



The Software Development Process

- **Analyze the Problem**

Figure out exactly the problem to be solved. Try to understand it as much as possible.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 4



The Software Development Process

- **Determine Specifications**

Describe exactly what your program will do.

- Don't worry about *how* the program will work, but *what* it will do.
- Includes describing the inputs, outputs, and how they relate to one another.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 5



The Software Development Process

- **Create a Design**

- Formulate the overall structure of the program.
- This is where the *how* of the program gets worked out.
- You choose or develop your own algorithm that meets the specifications.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 6

The Software Development Process

■ Implement the Design

- Translate the design into a computer language.
- In this course we will use Python.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 7

The Software Development Process

■ Test/Debug the Program

- Try out your program to see if it worked.
- If there are any errors (*bugs*), they need to be located and fixed. This process is called *debugging*.
- Your goal is to find errors, so try everything that might “break” your program!

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 8

The Software Development Process

■ Maintain the Program

- Continue developing the program in response to the needs of your users.
- In the real world, most programs are never completely finished – they evolve over time.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 9

Key Things – Be Able to Define and Use These Concepts



- Variable
 - A container to hold the current value of a certain type
 - A name to refer to a value used in a program
- Statement
- Assignment



Python Reserved Words



- Python keeps some words or identifiers for its own use
 - DON'T use them for your own work
 - You'll learn to use many of them as a programmer

```
and del from not while
as elif global or with
assert else if pass yield
break except import print
class exec in raise
continue finally is return
def for lambda try
```



Example Program: Temperature Converter



- Analysis – the temperature is given in Celsius, user wants it expressed in degrees Fahrenheit.
- Specification
 - Input – temperature in Celsius
 - Output – temperature in Fahrenheit
 - Output = $9/5(\text{input}) + 32$



Example Program: Temperature Converter

- Design
 - Input, Process, Output (IPO)
 - Prompt the user for input (Celsius temperature)
 - Process it to convert it to Fahrenheit using $F = 9/5(C) + 32$
 - Output the result by displaying it on the screen

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 13



Example Program: Temperature Converter

- Before we start coding, let's write a rough draft of the program in *pseudocode*
- Pseudocode is precise English that describes what a program does, step by step.
- Using pseudocode, we can concentrate on the algorithm rather than the programming language.

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 14



Example Program: Temperature Converter

- Pseudocode:
 - Input the temperature in degrees Celsius (call it celsius)
 - Calculate fahrenheit as $(9/5)*celsius+32$
 - Output fahrenheit
- Now we need to convert this to Python!

10/19/2009

Adapted from John Zelle, *Python Programming*, 1e 15



Elements of Programs

- Names
 - Names are given to variables (celsius, fahrenheit), modules (main, convert), etc.
 - These names are called *identifiers*
 - Every identifier must begin with a letter or underscore (“_”), followed by any sequence of letters, digits, or underscores.
 - Identifiers are case sensitive.

10/19/2009

Adapted from John Zelle, *Python Programming, 1e* 16



Elements of Programs

- Expressions
 - The fragments of code that produce or calculate new data values are called *expressions*.
 - *Literals* are used to represent a specific value, e.g. 3.9, 1, 1.0
 - Simple identifiers can also be expressions.

10/19/2009

Adapted from John Zelle, *Python Programming, 1e* 17
