Introduction to Computer Programming

Lecture 2
Input, Processing, Output
Topics

• Designing a Program
• How to use IDLE
• Comments
• Displaying Output with \texttt{print} Function
• Strings
• Variables
• Reading Input from the Keyboard
The more time designing, the less time debugging!
Gathering Requirements

Example: Write a Program to compute grades.

Before you do the design, all sorts of questions will pop into your head. Need to get details from the “customer”.

- Use weighted average?
- Grade on a scale?
- Letter or number grades?
- Can the teacher override the grades?

Have new questions all the time.

Have to do lots of documentation.

The more detailed you can be, the less you will have to change later.
Designing Your Algorithm

• Before starting to program, it’s wise to break down your problem into a sequence of small steps.
  – Input, Computation, Output

• The sequence of steps is called an Algorithm.

• There are several standard ways of documenting your algorithm design.
Flowchart for Computing Grade Average

- **Start**
- **Input Midterm Grade**
- **Input Final Grade**
- **Input Homework Grade**
- **Calculate Course Grade as the Average Of Input Grades**
- **Display Course Grade**
- **Stop**

**Ovals are terminals**

**Parallelograms are input/output**

**Rectangles are computations**
Pseudocode for Computing Grade Average

1. Input the midterm grade
2. Input the final grade
3. Input the homework grade
4. Calculate course grade as the average of the input grades
5. Display the final grade
Commenting your Code

• A comment is a note that a programmer puts in her program to explain what various lines of code are supposed to do.

• Comments are part of your program.

• It is very important to include good comments
  – Most of the time, programmers are modifying and reusing code written by other people.
  – Often you have to come back to your code you wrote a long time ago.

• Comments are completely ignored by the compiler – they are for people’s eyes only.
Commenting your code

- You should have a comment at the beginning of your program to describe what it does.
- You can also have more detailed comments later on to explain your logic.
  - End-of-line comments

- In Python, comments begin with the character `#`
  - The rest of the line is ignored.

- You must have comments in your homework
  - Include your name and assignment number.
Debugging: Types of Errors

• Syntax: Detected by the Compiler
  -- Example: print("Hello World")

• Runtime: Program Execution will Abort
  – Example: 10/0 cannot divide by 0!

• Logical: Produces wrong answer
  – Example: Average of 100, 90, 80 is printed as 130!

You must test your program on different inputs.
Debugging Tips

• Test as you go...
  – Run pieces of your program as you write them.

• Use a lot of print statements to verify intermediate results.
First Program: Echo

Start

Input Text

Echo Text 5 times

Stop
Review: Functions

• A Function is a piece of prewritten code that performs an operations
  – Examples: square root, area, draw, compute sales tax, anything you want
  – An Argument is the data given to the function as input
  – Functions can also return output data

Syntax:  

```
function_name(argument, argument, ..)

variable = function_name(argument, ...
```

Review: The print Function

• The **print** function displays output of the screen
  – It is *built-in* function, i.e. comes with Python
  – You will learn to write your own functions later
  – Argument: Data that is printed on the screen
• The print function can display more than one argument
  – Arguments are separated by commas
  – They are displayed in order, with a space between them
Review: Strings and String Literals

• A **String** is text data, i.e. a sequence of characters that is used as data
  – *String* is a data type

• A **String Literal** is text that appears in your program in quotes: “..” or ‘...’
  – It’s a specific String
  – The *data type* of a String Literal is String.
Review: Reading Input from the keyboard

• Many programs need to read input from the user

• Built-in `input` function reads input from keyboard
  
  – Returns the data as a string
  
  – Format: `variable = input(prompt)`
    
    • `prompt` is typically a string instructing user to enter a value
  
  – Does not automatically display a space after the prompt
Review: Variables

• A **Variable** is used to save or “remember” a data value
  – Needed to store and access data in main memory

• An **Assignment Statement** gives a variable a data value. It is not an equation!
  – Syntax:  `variable_name = value`
  – Example:  `firstName = “Mary”`
  – This means:.firstName gets the value “Mary”
Review: Variables (continued)

- In assignment statement, the variable must be on the left hand side:
  - **WRONG:** “Mary” = firstName

- A variable can be passed as an argument in a function --- no quotes!
  - **WRONG:** print(“firstName”)

- A variable must be assigned a value before you can use it
  - **WRONG:** print(someNewVariable)
Variable Naming Rules

• Python keywords cannot be used as variable names
• A variable name cannot contain spaces
• The first character must be a letter or underscore (_)
• Other characters can be letters, numbers, or underscore(_)
• Uppercase and lowercase are distinct
  – Mary is not the same as mary

By convention, variables begin with a lower case letter
Python Keywords Cannot be Used as Variable Names!

and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield
Give your variables meaning names that reflect their purpose

• X and Y are not good variable names
• Variable names for temperature conversion:
  – Numbering variables, not very meaningful
    temperature1, temperature2
  – Multiple words, hard to read:
    farenheittemperature, centigradetemperature
  – Better:
    farenheit_temperature, centigrade_temperature
    or
    farenheitTemperature, centigradeTemperature
Short Quiz
Legal or Illegal Variable Name?

76trombones
give_me_$$$
Lottsa_luck
_firstName
class
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones  Illegal!
give_me$$$$
Lottsa_luck
_firstName
class
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones  Illegal!
give_me_$$$  Illegal!
Lottsa_luck
(firstName
_class
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones   Illegal!
give_me_$$$   Illegal!
Lottsa_luck    Legal!
_firstName     
class
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones    Illegal!
give_me_$$$     Illegal!
Lottsa_luck     Legal!
_firstName      Legal!
class
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones  Illegal!
give_me_$$$$  Illegal!
Lottsa_luck  Legal!
_firstName  Legal!
Class  Illegal!
Class
Short Quiz
Legal or Illegal Variable Name?

76trombones    Illegal!
give_me_$$$    Illegal!
Lottsa_luck    Legal!
_firstName     Legal!
Class           Illegal!
Class           Legal!