VI

Vishwakarma Institute of Technology

Python Programming

Prof. Sujit M. Deokar,
Department of Engineering & Sciences Humanities
Vishwakarma Institute of Technology, Pune

Introduction

What Is a Program?

- Usually, one or more algorithms written in a programming language that can be translated to run on a real machine
- We sometimes call programs ***software***

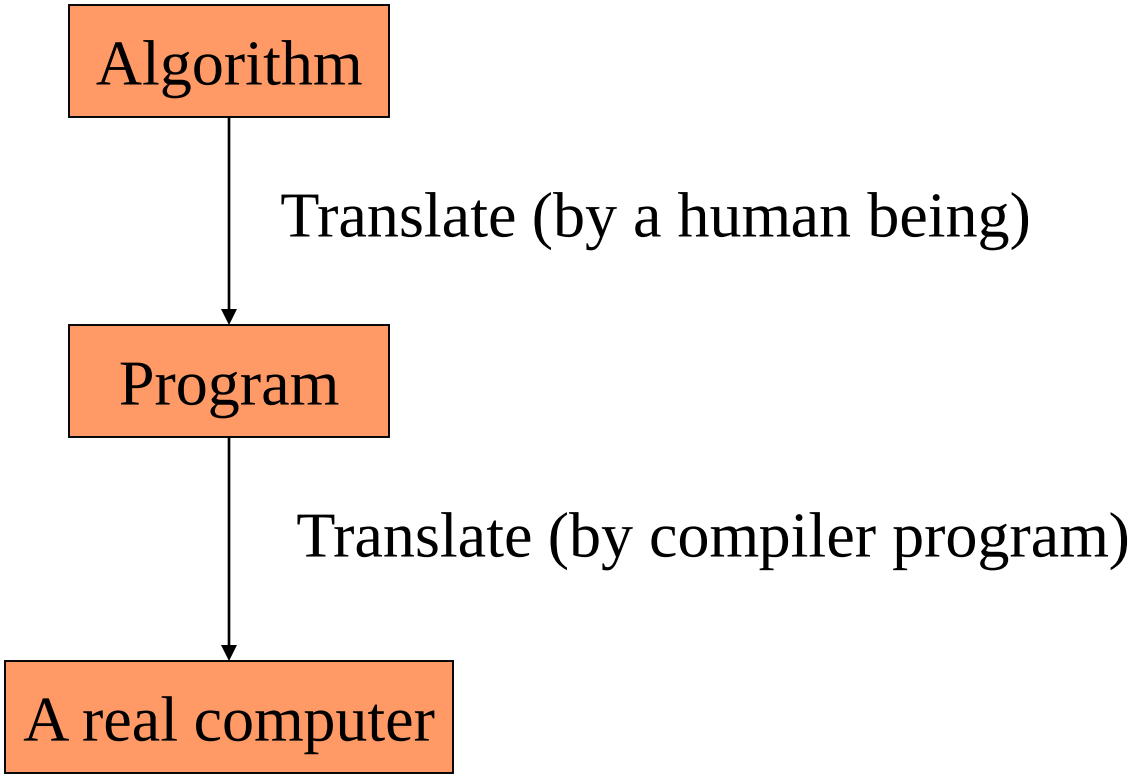
What Is a Programming Language?

- A programming language is somewhat like a natural language, but with a very limited set of statements and strict syntax rules.
- Has statements to implement sequential, conditional and iterative processing - algorithms
- Examples: FORTRAN, COBOL, Lisp, Basic, Pascal, C, C^{++}, Java, C#, Python, …

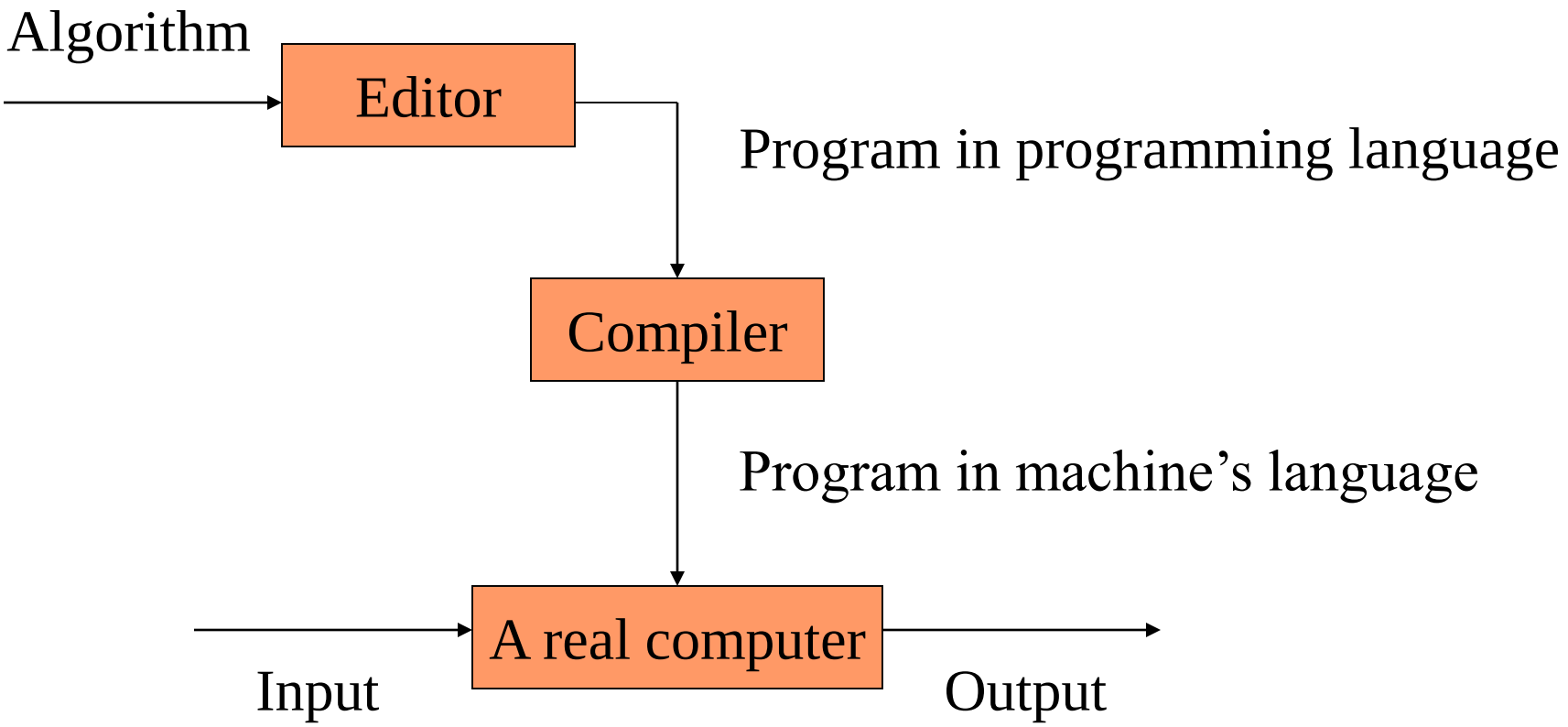
Compiler

- A *compiler* is a program that converts a program written in a programming language into a program in the native language, called *machine language*, of the machine that is to execute the program.

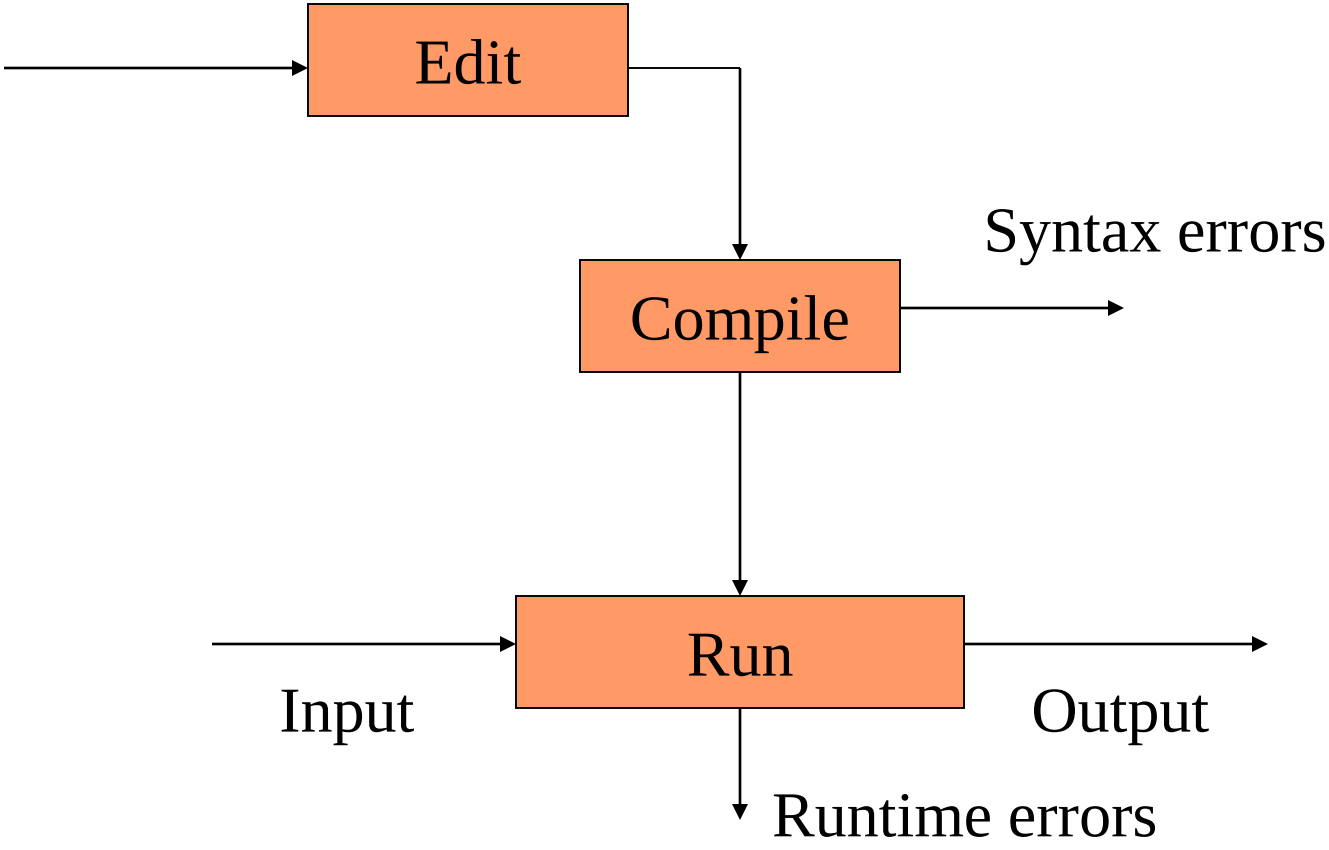
From Algorithms to Hardware (with compiler)



The Program Development Process (Data Flow)



The Program Development Process (Control Flow)



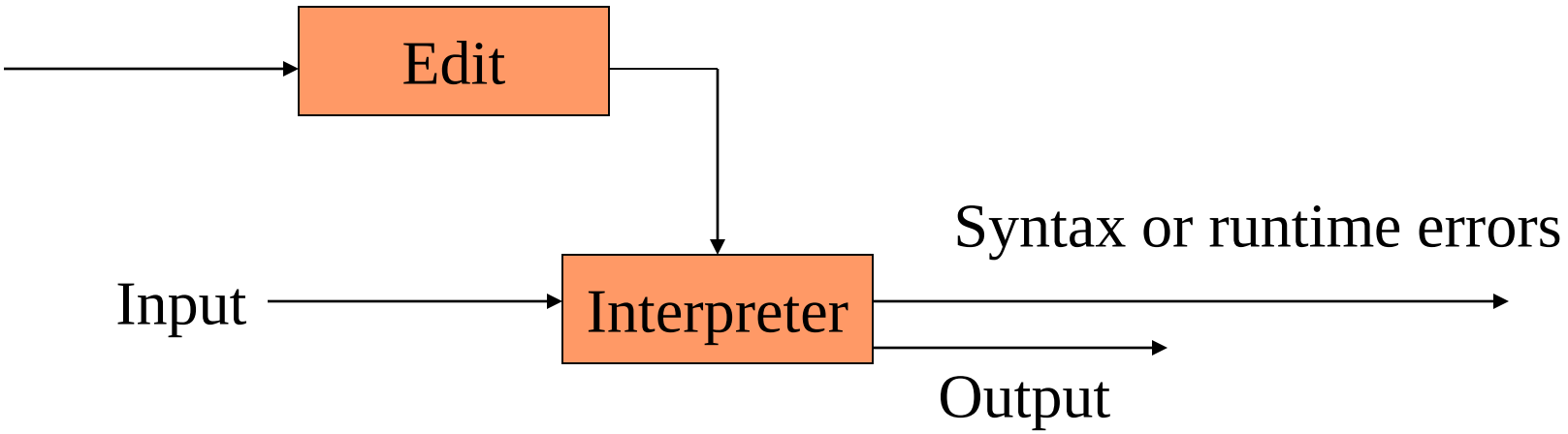
Three kinds of errors

- *Syntax error* : Some statement in the program is not a legal statement in the language.
- *Runtime error* : An error occurs while the program is executing, causing the program to terminate (divide by zero, etc.)
- *Logic error* : The program executes to completion, but gives incorrect results.

Interpreter

- An alternative to a compiler is a program called an *interpreter*. Rather than convert our program to the language of the computer, the interpreter takes our program one statement at a time and executes a corresponding set of machine instructions.

Interpreter



Python

- Python is a real-world, production language that is freely available for most computers.

  http:www.python.org

- If you want a copy of Python to use with this course, go to

  http://code.google.com/p/mediacomp-jes/ .

  We are using JES (Jython Environment for Students) which has a lot of special multimedia functionality.

- Note: Our textbook covers a limited amount of Python. There are many excellent online tutorials. For example, see

  http://en.wikibooks.org/wiki/Non-Programmer's_Tutorial_for_Python/Contents

Python

- Python uses an interpreter. Not only can we write complete programs, we can work with the interpreter in a statement by statement mode enabling us to experiment quite easily.
- Python is especially good for our purposes in that it does not have a lot of “overhead” before getting started.
- It is easy to jump in and experiment with Python in an interactive fashion.

Language terminology

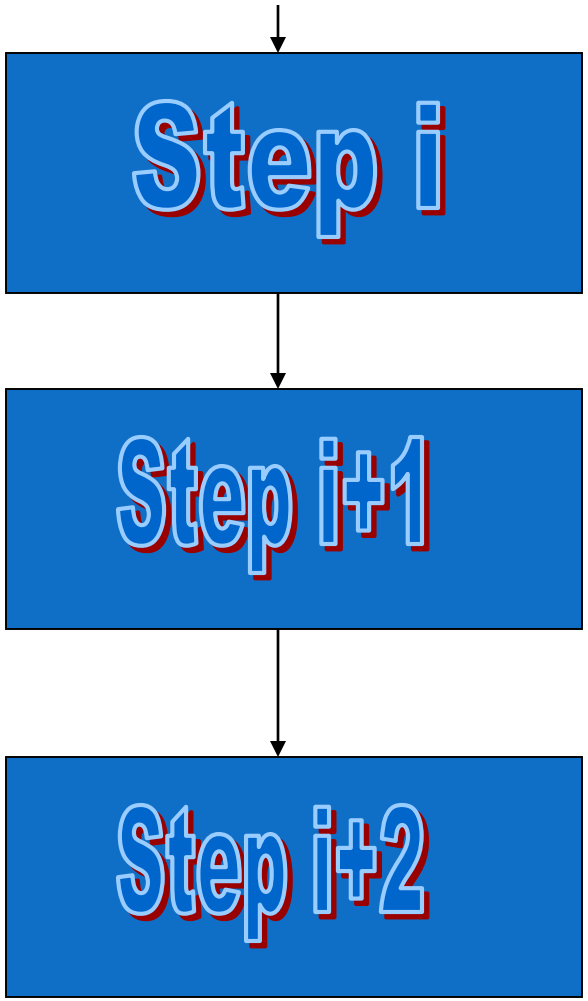
- Syntax: The formal rules for legal statements in the language.
- Semantics: The meaning of the statements - what happens when the statement is executed.

Three major control constructs of programming
(Execution flow of instructions)

- Sequential: Simply do steps one after the other in order they are listed.
- Conditional: Decide which statement to do next based on some true/false test.
- Iterative: A set of statements is repeated over and over until some condition is met.

Sequential Operations

"Atomic"



- Input
- Computation
- Output

The Big Plan

- We want to get some experience of programming simple algorithms in a real programming language. This gives us an understanding of how software is written and allows us to test our algorithms to see if they work.
- We'll first work with programs where the variables have numbers as values.
- Later we'll work with programs dealing with pictures and sound.
- In lab we'll work with some simple statements and small programs.

The Basic Pattern

- Most of our programs will use the basic pattern of
  - Get some user input
  - Perform some algorithm on the input
  - Provide results as output

Identifiers

- *Identifiers* are names of various program elements in the code that uniquely identify the elements. They are the names of things like variables or functions to be performed. They're specified by the programmer and should have names that indicate their purpose.

- In Python, identifiers
  - Are made of letters, digits and underscores
  - Must begin with a letter or an underscore
  - Examples: temperature, myPayrate, score2

Keywords

- *Keywords* are reserved words that have special meaning in the Python language. Because they are reserved, they can not be used as identifiers. Examples of keywords are *if*, *while*, *class*, *import*.

Variables in Python

- A variable has
  - A name – identifier
  - A data type - int, float, str, etc.
  - Storage space sufficient for the type.

Numeric Data Types

- int

  This type is for whole numbers, positive or negative. Examples: 23, -1756

- float

  This type is for numbers with possible fraction parts. Examples: 23.0, -14.561

Integer operators

The operations for integers are:

+ for addition

- for subtraction

* for multiplication

/ for integer division: The result of 14/5 is 2

% for remainder: The result of 14 % 5 is 4

- *, /, % take precedence over +, -
  x + y * z will do y*z first

- Use parentheses to dictate order you want.
  (x+y) * z will do x+y first.

Integer Expressions

- Integer expressions are formed using
  - Integer Constants
  - Integer Variables
  - Integer Operators
  - Parentheses

Python Assignment Statements

- In Python, = is called the *assignment operator* and an *assignment statement* has the form

  <variable> = <expression>

- Here
  - <variable> would be replaced by an actual variable
  - <expression> would be replaced by an expression

- Python: age = 19

Python Assignment Statement

- Syntax: <variable> = <expression>
  - Note that variable is on left

- Semantics:

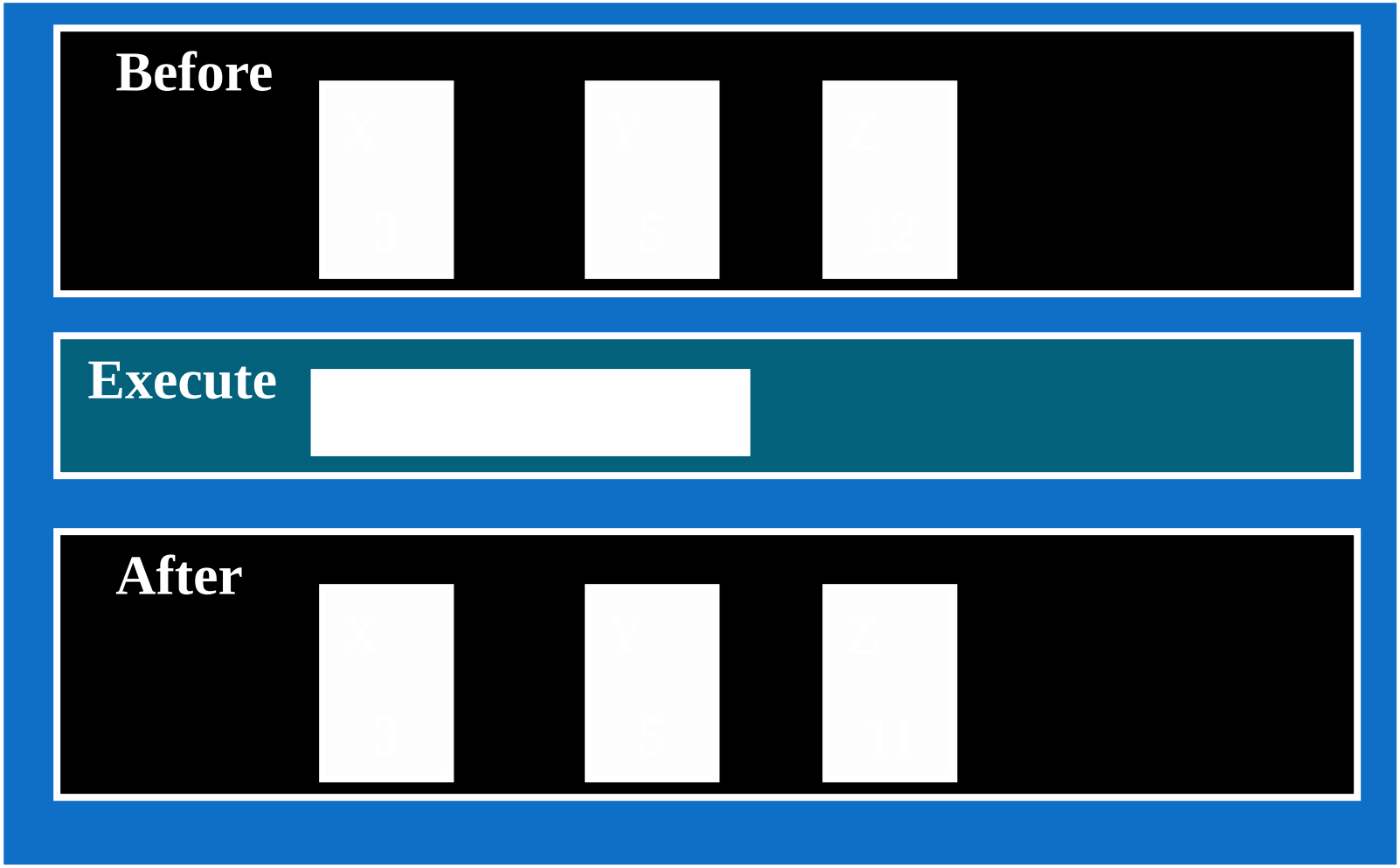
  Compute value of expression

  Store this as new value of the variable

- Example: Pay = PayRate * Hours

10	40	400
Payrate	Hours	Pay

Assignment Example



VI

Vishwakarma Institute of Technology

Python Session

JES - Jython Environment for Students - Untitled

File Edit Watcher MediaTools JES Functions Window Layout Help

Load Program Watcher Stop

```
>>> hoursWorked
The error was:hoursWorked
Name not found globally.
A local or global name could not be found. You need to define the function or variable before
you try to use it in any way.
>>> hoursWorked = 15
>>> hoursWorked
15
>>> payRate = 20
>>> totalPay = payRate * hoursWorked
>>> totalPay
300
>>> type(totalPay)
<type 'int'>
>>>
```

Python Session

```
>>>
>>> 14/5
2
>>> 14 % 5
4
>>> 12-3  * 4
0
>>> (12-3) * 4
36
>>>
```

What about floats?

- When computing with floats, / will indicate regular division with fractional results.
- Constants will have a decimal point.
- 14.0/5.0 will give 2.8 while 14/5 gives 2.

Comments

- Often we want to put some documentation in our program. These are comments for explanation, but not executed by the computer.

- If we have # anywhere on a line, everything following this on the line is a comment – ignored

Numerical Input

- To get numerical input from the user, we use an assignment statement of the form

  <variable> = input(<prompt>)

- Here
  - <prompt> would be replaced by a prompt for the user inside quotation marks
  - If there is no prompt, the parentheses are still needed
- Semantics
  - The prompt will be displayed
  - User enters number
  - Value entered is stored as the value of the variable

Print Statement

- For output we use statements of the form

  print <expression>

- Semantics
  - Value of expression is computed
  - This value is displayed

- Several expressions can be printed – separate them by commas

Example - Fahrenheit to Centigrade

- We want to convert a Fahrenheit temperature to Centigrade.
- The formula is $C = (F - 32) \times 5/9$
- We use type float for the temperatures.

Python Session

```
>>> fahrenheit = input("Enter fahrenheit temperature: ")
Enter fahrenheit temperature: 200
>>> centigrade = (fahrenheit - 32) * (5/9)
>>> print centigrade
0
>>> centigrade = (fahrenheit - 32) * (5.0/9.0)
>>> print centigrade
93.33333333333334
>>>
```

If we figure this out once, let's remember what we did!