ITD62-123 COMPUTER PROGRAMMING

IMI62-122 FUNDAMENTAL OF COMPUTER PROGRAMMING

Theerat Saichoo & Yunyong Punsawad
School of Informatics, Walailak University

Chapter 1

Introduction to computer programming

PART 1: Intro

Topics

- Getting start in computer programming
- Overview of Python
- Steps for programming
- 5 Ways to learn programming faster

Getting start in computer programming

- Coding to command somethings
- 2 types of programming
  - Structure programming
  - Object-oriented programming
- How to choose programming language?
  - Basic of knowledge
  - Resource
  - Community
  - Framework
  - Cost

Getting start in computer programming

- Learning programming independent on language.
- Thinking about logic more than syntax.
- Practice is the best way to learn programming.
- Method
  - Class Activity – coding on the paper
  - Laboratory Practice – coding on the computer
- Don’t worry about syntax but solves the problems first.

Getting start in computer programming

- Idea or a problem that you wish to solve.
- What do you want the software to do?
- Who will be using it?
- What system will users be running it on?

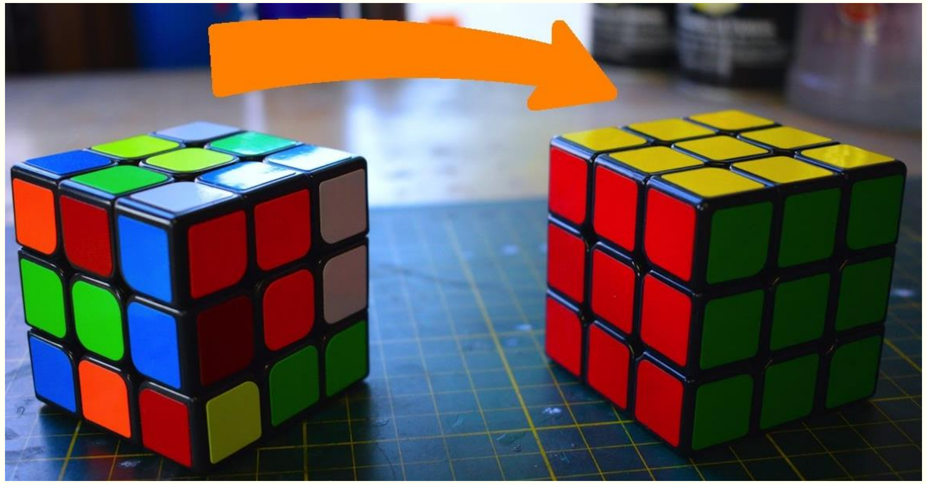
Overview of Python

- The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum.
- Python reached version 1.0 in January 1994.
- Python 2.0 released October 2000.
- Python 3.0 was released on December 3, 2008.
- Python is similar in style to pseudocode.

Steps for programming

- 4 steps for programming
  1. Systematic thinking

     Solves any problem by make an algorithm first.



Steps for programming

- 4 steps for programming

2. Coding a programming

   Type the source code into the compiler.

```
#program print
#filename hello.py

print("Hello world")
```

Steps for programming

- 4 steps for programming

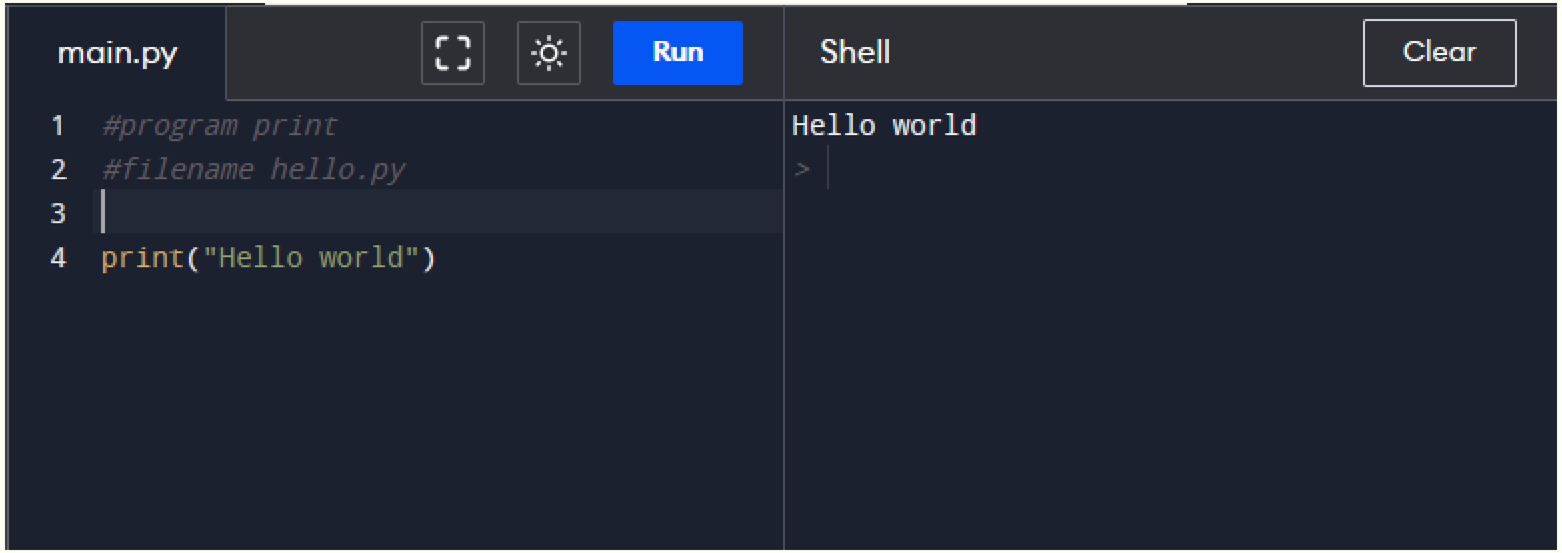
  3. Compiling
     - The source code written in source file is the human readable source for your program.
     - It needs to be "compiled" into machine language so that your CPU can actually execute the program as per the instructions given.

Steps for programming

- 4 steps for programming

  4. Executing.

     - Run a program



```
#program print
#filename hello.py

print("Hello world")
```

```
Hello world
>
```

5 Ways to learn programming faster

1. Look at the example code
   - Try to understand every example.
   - What it did?
2. Don’t just read example code only
   - Please run it into a compiler by
     - Typing by yourself, you will understand the details of syntax.
     - Shouldn’t COPY & PASTE.
     - Compile
     - Run
     - Change

5 Ways to learn programming faster

3. Write your own code as soon as possible.
   - Re-implement the examples from the book.
   - Try to do without looking back at the sample code.
4. Learn to use a Debugger
   - Step line by line through a piece of code
     - See the value of variables.
   - Help you quickly answer questions about what your coding is doing.
   - Don’t make life hard by unwilling to use debugger.

5 Ways to learn programming faster

5. Seek out more sources.
    - Look for alternative explanations
        - Internet
        - Good books
        - Expert
        - Teacher
        - Good friends
    - Ask smart
        - Don’t say “I don’t understand, pls explain”.
        - But say “How your thinking”.

Part 2: Variable and Data Type

Topics

- Variables
- Data types
- Creating Variables and Assigning Values

Variables

- There are just a couple of rules to follow when naming your variables.
    - Variable names can contain letters, numbers, and the underscore.
    - Variable names *cannot* contain spaces.
    - Variable names *cannot* start with a number.
    - Case matters — for instance, temp and Temp are different.
- It helps make your program more understandable if you choose names that are descriptive, but *not* so long that they clutter up your program.

Variables

- 33 keywords which are reserved

False	None	True	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	if	import
in	is	lambda	nonlocal	not
or	pass	raise	return	try
while	with	yield		

Variables

- Examples of acceptable identifiers:

mohd zara abc move_name

a_123 myname50 _temp j

a23b9 retVal

✓

time@wu get$next

If for

✗

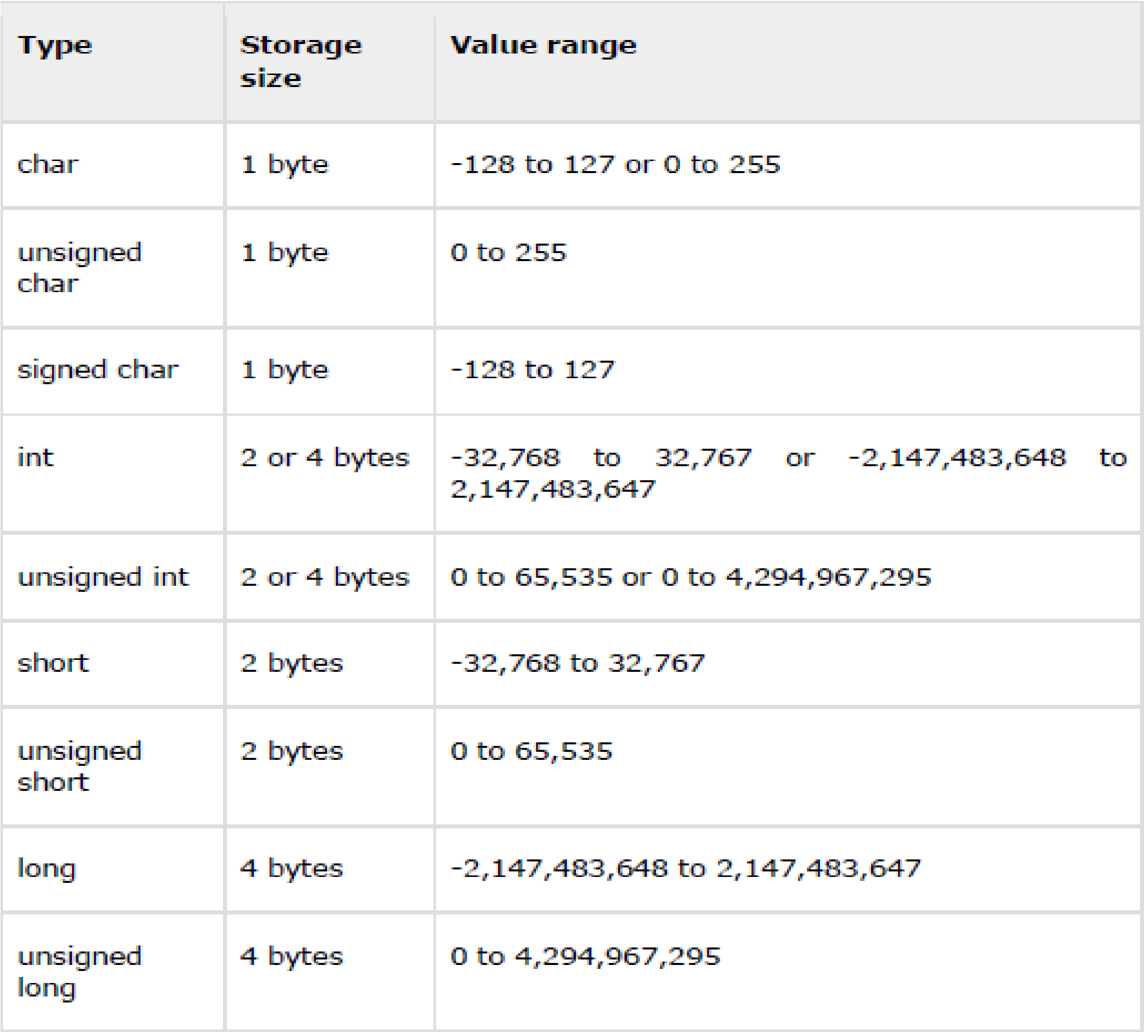
Data types

- Data types in Python
  - Integer
  - Floating point
  - String

Data types

- Integer 32-bit
  - …, -3, -2, -1, 0, 1, 2, 3, …

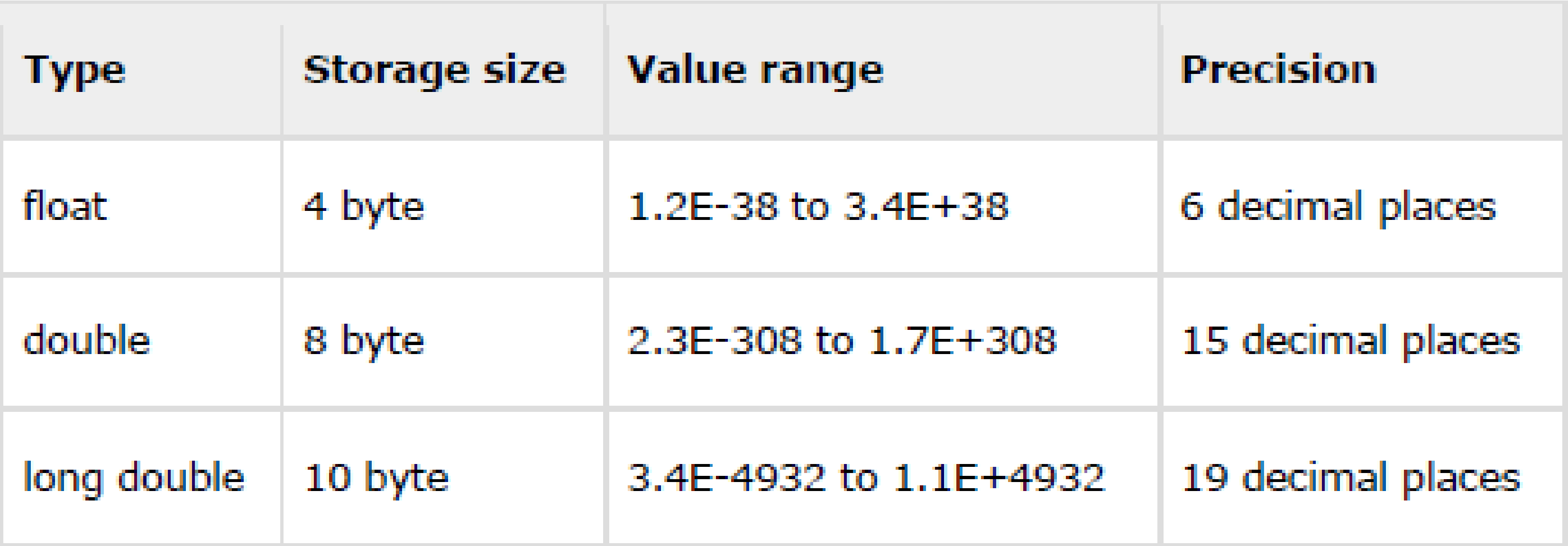
- price = 199



Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

Data types

- Floating point
  - 3.25 , 100.0081, 2.44E+300
  - gpax = 4.00



Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

Data types

- String
  - A group of characters or text
  - Be enclosed in double quotes (“ ”) or (‘ ’)
  - Ex – “Computer Programming” or ‘Computer Programming’

- firstName= “Theerat”
- lastName = ‘Saichoo’

Creating Variables and Assigning Values

- number = 0
- roll_width = 1.4
- price_per_metre = 5
- filename = 'data.txt'
- trace = False
- sentence = "this is a whole lot of nothing"

Class Activity 1 : Variables & Constants Definition

- Naming variables or constants and assign data type
  - iPhone Price
  - Product Name
  - First name
  - Last name
  - School name
  - GPAX
  - Student ID
  - Bus fare
  - Grade
  - Pi
  - Circle Radius
  - Average value
  - Number of persons
  - Interest
  - Address
  - Postcode
  - Room number
  - Date
  - Time
  - Telephone number

THE END