

Lab 02

Program Development

Program Development Cycle

1. Problem Analyses
2. Program Design
3. Coding
- ~~4. Debugging~~
- ~~5. (Input) Validation~~
- ~~6. Documentation~~
- ~~7. Program Maintenance~~
8. Input-Output and Variable

- Note that the steps here are not necessarily once-and-done
- You may have to visit earlier steps later one
- Some step (such as documentation), you will have to perform throughout the development cycle

Problem Analyses

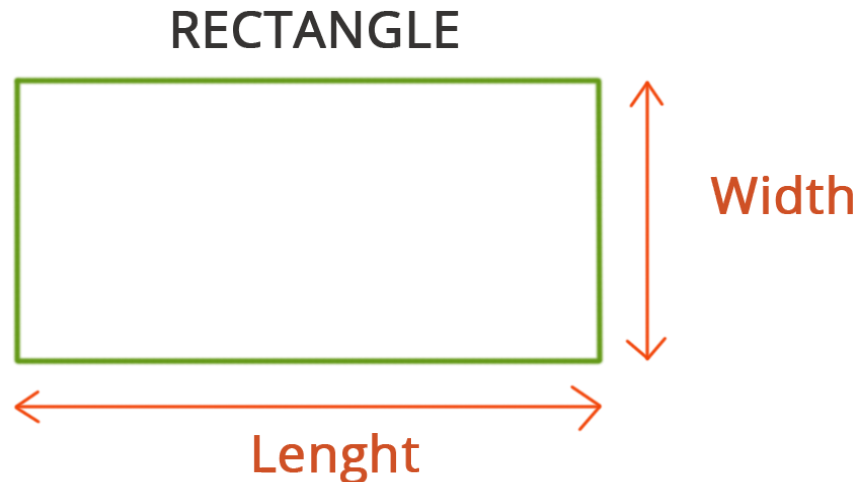
- Taking the problem, usually written in human languages, and:
 - Identify components of the problem (input, output, process)
 - Identify processing steps in solving the problem
- Analysis tasks:

Output:	Input:	Process:
The result(s) we want	Data we start with	Steps to transform input data into desired output

1. Problem Analyses – Example #01

- Problem:

- “Find the area of the specified rectangle, given its length and width.”



Area of rectangle = length x width

1. Problem Analyses – Example #01

- Problem:

- “Find the area of the specified rectangle, given its length and width.”

- Analysis Result

Output:	Input:	Process:
The area of the rectangle	Length and width of the rectangle	length * width
-Which Unit? cm ² , m ² , inch ² , acre?	-Which Unit? Same as desired output?	-Unit conversion?

2. Program Design

- Lay out the structure of the program
 - Step-by-step working of the program
- Somewhat like architectural plan
 - The building isn't built yet, but you know what it looks like.
- You will use design tools
 - Algorithm Description
 - Pseudocode
 - Flowchart

2. Program Design – Algorithm Description

- Describing program's working in human language
- Describe the process in clear, unambiguous steps
- Example: finding the area of the rectangle

1. Receive the length of the rectangle
2. Receive the width of the rectangle
3. Calculate the area of the rectangle using the following formula:
 - $Area = Length \times Width$
4. Display the resulting area of the rectangle

2. Program Design – Pseudocode

- Describe the process in programming-language like description
 - Make sure everyone in the team agree on the pseudocode syntax!
- Specify variables for storing
 - Input data - length (cm) and width (cm)
 - Output result(s) - area (cm²)
 - (If needed) Intermediate result(s)

2. Program Design – Pseudocode Example

- Example: Finding area of a rectangle

START

READ width

READ height

CALCULATE Area = width * height

DISPLAY Area

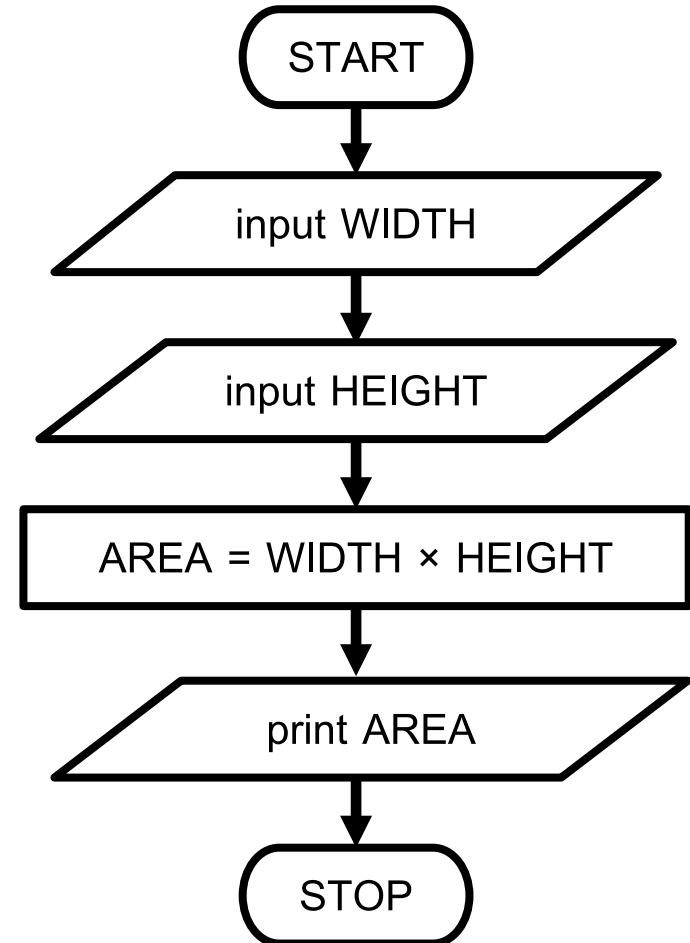
STOP

* is used as multiplication in most
programming language

2. Program Design – Flowchart

- Describe the process by symbols
- Direction line (\rightarrow) will show the order of execution among the steps (blocks)
- Easy to see the overall working of the program

Example: Calculating area of a rectangle

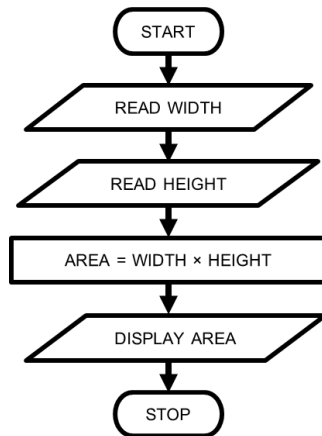
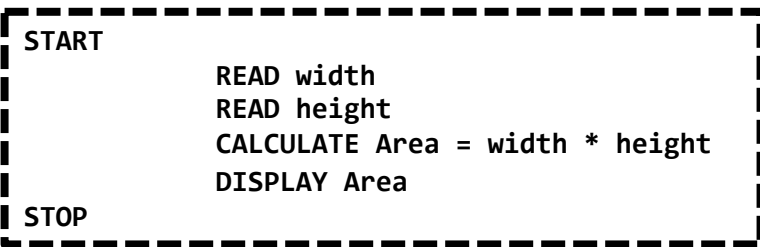


3. Coding

- Writing the program using programming language (such as Python)
- Translate the program design into computer language code
- The code then can be translated into working computer program

3. Coding (cont.)

- Example: translate flowchart for calculating rectangle area into Python code



```
width = int(input())
length = int(input())
area = width * length
print(area)
```

- The Python interpreter used in this class is CPython (<https://www.python.org/>), which is the reference implementation created by Guido van Rossum, the creator of the Python language. CPython is the most widely used implementation.
- In addition to CPython, there are other Python interpreters available
 - Jython, written in Java for the JVM,
 - PyPy, written in RPython
 - IronPython, written in C#
- The CPython interpreter has two modes: the Command-line Mode and the Script Mode.



Interactive Mode

- Python offers an interactive mode for interactive coding and immediate feedback.
- Interactive mode is commonly used for testing small code snippets, exploring Python features, and learning the language interactively.
- Online Version of Python Interactive Interpreter
<http://repl.it/languages/Python>

```
$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more
information.
>>> print(1 + 1)
2
```

Script Mode

```
$ python3 test.py  
2
```

- Python offers a script mode for executing Python programs.
- Script mode allows the execution of multiple Python statements or a complete program.
- In script mode, Python programs are typically stored in plain text files with a .py extension.
- To run a Python script, open the terminal or command prompt, navigate to the script's directory, and enter the command "python script_name.py".
- Script mode is commonly used for writing and running standalone Python programs or scripts.

Installing Python

- Version 3.8.10 (Same version to the Grader)
- Integrated Development Environment (IDE)
 - Python provided built-in IDE named IDLE
 - Or you can use another Text Editor
 - Visual Studio Code and PyCharm are recommended

Grader

- We will use an automated grader for programming assignment
- Username: (your Student ID) and passwords are sent to



- Grader Link
 - <https://cmu.to/grader204101>

Grader Example 01

“Hello World”

Grader Example 01: “Hello World”

- Create and Upload to HW01_0 on the course grader:

Hello World

```
01 #!/usr/bin/env python3
02 # first_name last_name
03 # 6XXXXXXXXX
04 # HW01_0
05 # 204101 Sec 701
06
07 print("Hello World")
```

must have for all submission

Input-output Statements

- For a program to be able to interact with the user, the program need a way to:
 - Receive data from the user (input), and
 - Display result to user (output)
- We will start with:
 - **Output:** `print()` function
 - **Input:** `input()` function

print()

- `print()` will display its arguments (values inside the parentheses) on interactive mode windows.

Code:

```
print ("Welcome ")  
print ("to python" )
```

Output:

```
Welcome  
to python
```

print()

- `print()` without the automatically added newline character, you need to set the `end` parameter of the `print()` function to an empty string (`" "`)

Code:

```
print ("Welcome ", end="")  
print ("to python" )
```

Output:

```
Welcome to python
```

print()

You can print multiple strings/variables at the same time by separating them with comma (.). The texts will be concatenated when printed.

Code:

```
name = 'ICDI'  
ver = 3.9  
print ("Hello ! " , name , "Welcome to python" , ver)
```

Output:

```
Hello ! ICDI Welcome to python 3.9
```

Grader Example 02

Compute a times b

Variables

- Variables are named containers used to store data in a program.
- Each variable has a name and a value.
- Variables can hold different types of data, such as numbers, text, or Boolean values.

Variables

- To create a variable, you use an assignment statement, such as `variable_name = value`



05	<code>pi = 3.14</code>
06	<code>radius = 11</code>
07	<code>area = pi * radius * radius</code>

- The value of a variable can be changed throughout the program.
- Variables are used to store and manipulate data, perform calculations, and keep track of information.

Grader Example 02

- Compute a times b
- click at instruction on grader and download skeleton file



Task	Name
Example_01	Print "Hello World"
Example_02 	Compute a times b
Example_03 	Compute a times b from user inputs

0%

</> Compute a times b

Example_02

Task Details

Type	Batch
Time limit	1 second
Memory limit	16 MiB

Submission


 Submit your code

Statement

No statement available

Attachments

EX02-skeleton.py

 Python script 63 B

1. download the skeleton code
2. write your solution on code and save
3. upload your code to check for score

Grader Example 03

Compute a times b from user
inputs

8.2 input()

- `input()` function will prompt user (with the message inside the parentheses) to enter data. User will type out data, then press enter.
- Received input will be of string type, and we need a variable to hold the data (function return)

Example:

```
str = input("Enter your input: ")  
print ("Received input is:", str)
```

Result (user input in blue):

```
Enter your input: Hello Python  
Received input is: Hello Python
```

- From the example, “Hello Python” will be stored in variable `str`

8.2 input() (cont.)

- Since the input will be string (text), it cannot be used in calculation right away
- We will need to convert the input into appropriate type first (int or float)

Example:

```
# prompt user for input and store input in variable in_string_1
in_string_1 = input("Input integer number: ")

# convert text input into integer and store on variable int_val
int_val = int(in_string_1)

# get next user input and store into variable in_string_2
in_string_2 = input("Input float number: ")

# convert to float, then store it on variable float_val
float_val = float(in_string_2)
```

8.2 input() (cont.)

- Input and convert type (It behaves the same way as the code in the previous slide, but this is a shorter version)

Example:

```
# prompt user for input, convert to int then store in var_1
int_val = int(input("Input integer number: "))
```

```
# get another input and convert to float, then store in var_2
float_val = float(input("Input float number: "))
```


8.2 input() (cont.)

- Warning: be careful about input data type.

Example:


```
>>> inp = int(input("input an integer: "))
input an integer: 1.2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '1.2'
>>>
```

- If the user input 1.2, it will read and put into inp with problem. BUT when Python try to convert it to integer, an error will occur (since 1.2 is not an integer)

Grader Example 03

- Compute a times b from user inputs click at instruction on grader and download skeleton file





Task	Name
Example_01	Print "Hello World"
Example_02 	Compute a times b
Example_03 	Compute a times b from user inputs

0%

Lab Assignment

- Complete 3 tasks on course grader as followed

 204101_66_inter

Task	Name	Status	Public Score	Time limit	Memory limit
Example_01	Print "Hello World"	Evaluated	1 / 1	1 second	16 MiB
Example_02 	Compute a times b	Evaluated	1 / 1	1 second	16 MiB
Example_03 	Compute a times b from user inputs	Evaluated	5 / 5	1 second	16 MiB

100%

Lab Exercise

One Last Exercise For Lab 02 (Not graded)

(Lab Exercise) Write a program that prompts the user to enter their name, surname, and age. The program should then output the provided information, along with the age at which the user will graduate (assuming a standard four-year duration). **If you are done, show your code to the instructor and/or the TA for review.**

Input (user input in red)

Input name: Chalee
Input surname: Buddee
Input age: 18



Output

My name is Chalee
And my surname is Buddee
Now I'm 18 years old
I'll graduate when I'm 22