ISYS90088
Introduction to Application Development

School of Computing and Information Systems
University of Melbourne
Semester 2 , 2018

Dr Antonette Mendoza

* Some slides are adapted from Fundamentals of Python by Kenneth A. Lambert and dept. resources

Lecture Agenda

- Introduction
  - Objectives
  - Staff ; Learning & Assessment
- What is a computer, and how do we talk to it?
- Python
- Grok
- Python Basics - print, input, strings

ISYS90088 sem 2 2018

About me (Antonette)

- PhD in Computer Science – UoM
- 17 years @ UoM teaching
- Three Dept. and MSE Teaching Awards
- University of Melbourne Edward Brown Teaching Award (2017)
- Australian Award for University Teaching award (2017)
- Computing Research and Education teaching award (2018)
- Coordinator – Tutor development Program in the MSE; New academic teaching development program in MSE; Graduate research coordinator in CIS
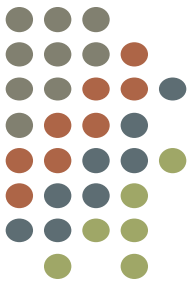
About me (Antonette)

- ## Research focus:
  - Expertise lies in the area of process modelling, design and development of emerging technologies that support positive computing, adoption and longer-term use.
  - Interests extend to exploring the impact of wider range of technologies that include mobile applications and social media platforms on society including health care, e-learning and telework.

- ## Grants:
  - Australian Research Council grant 2016 – 2018
    - "Catering for individuals' emotions in technology development"
  - CRC Low Carbon Living grant 2014 – 2017
    - Increasing knowledge and motivating collaborative action on low carbon living through team-based and game-based mobile learning' .
  - Other grants include: Learning and teaching grant 2015-2016; predictive tool for depression care NHMRC grant 2014 – 2015; aged care touch frame project; VVP project for industry client;
  - iPad app for assessment and feedback "Rapid Feedback" on iTunes in 2016.

About me (Antonette)

Contact details:

- Office: Room 6.03 Doug McDonnell Building
- Email: mendozaa@unimelb.edu.au
- Consultation: By appointment (email me and I will give you a time to meet)

Setting Expectations

What is Programming?

What is a Software development life cycle?

Is it important?

What do you expect to learn from this subject?

What we will be studying

- Fundamental concepts and models of application development

- Students who have no background in application development or programming.

- Learn about the software development lifecycle, program design, data structures, problem solving, programming logic, implementation considerations, testing and enterprise level applications.

Objectives

- Learn to use primitive data types and data structures

- Understand basic programming concepts

Assignment Project Exam Help

- Write simple applications that relate to a specific domain

https://powcoder.com

Add WeChat powcoder

- Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.

- Test applications

Learning and Assessment

- Lectures, Labs

Assignment Project Exam Help

- Assessment

https://powcoder.com

- Textbook & references Add WeChat powcoder

Learning and Assessment

Lectures: Tuesday 2.15pm – 4.15pm in Redmond Barry (Lyle Theatre)

Labs: 8 scheduled labs across the week – attend any one!

Workshop 1	Tuesday	18:15	20:15	Alan Gilbert-111
Workshop 2	Wednesday	17:15	19:15	Alan Gilbert-111
Workshop 3	Wednesday	19:15	21:15	Alan Gilbert-111
Workshop 4	Wednesday	16:15	18:15	Engineering B-102
Workshop 5	Thursday	19:15	21:15	Alan Gilbert-111
Workshop 6	Thursday	17:15	19:15	Elec. Engineering-121
Workshop 7	Thursday	16:15	18:15	PAR-Engineering B-102
Workshop 8	Friday	17:15	19:15	PAR-Alan Gilbert-111

Learning and Assessment

Tentative Schedule:

- Week 01: Introduction
- No workshops in week 01.

Assignment Project Exam Help

- Week02: Software development, data types and expressions
  https://powcoder.com
- Worksheet 1 – simple print and input stmt + how to work with grok + idle download

  Add WeChat powcoder

- Week03: math function + Conditions – for /while/if statement
- Worksheet 2 - data types, expressions

- Week04: Strings/sequences + math functions + Release ass 1 (10%)
- Worksheet 3 - related to if/for/while statement

Learning and Assessment

Tentative Schedule:

- Week05: complete sequences+ Functions
- Worksheet 4 – sequences and conditions + assignment 1

Assignment Project Exam Help

- Week06: Complete Functions – submit assignment 1

https://powcoder.com

- Worksheet 5 - functions & ass 1

Add WeChat powcoder

- Week07: Mid-semester test (10%)
- Worksheet 6 - Review so far!

- Week08: Iteration – for loops + while
- Worksheet 7 - iteration

Learning and Assessment

Tentative Schedule:

- Week09: complete + lists _ release assignment 2 (20%)
- Worksheet 8 – feedback mid semester test + work on assignment 2

- Week10: lists/dictionary
- Worksheet 9 - list/dictionary + assignment 2

- Week11: advanced functions & File I/O - submit assignment 2
- Worksheet 10 – advanced functions

- Week12: Any outstanding + exam revision (60%)
- Worksheet 11 – practice questions and/or catch-up

Learning and Assessment

Assessment:

Assignment Project Exam Help

- Individual Assignment 1 (10%) due in Week 6
- One mid-semester test (10%) in Week 7
- Individual Assignment 2 (20%) due in Week 11
- Exam (60%) – end of semester (3hr and will be a hurdle)

https://powcoder.com

Add WeChat powcoder

Learning and Assessment

Text books/references:

Assignment Project Exam Help

- You may use any text book and references.

- Some books/references https://powcoder.com

  ➤ Fundamentals of Python – Kenneth A. Lambert

  Add WeChat powcoder
  ➤ There are good online resources (free)

  ➤ Check out: *www.python.org*

LMS

- University's "Learning Management System"

- http://www.lms.unimelb.edu.au/

- Use your university email id and password

A word about the LMS & other environments

- We will post all code from lectures (other than snippets from the \console") on the LMS after each lecture;

  - ➢ It is a good idea to look back over the code to ensure you fully understand it and play around with it yourself

  - ➢ Make sure to attend Tutorials and labs

- You will work on GROK environment for your labs and assignments

Academic Honesty

➢ In accordance with the University's Academic Honest and Plagiarism Policy (which you should familiarize yourself with!):

Assignment Project Exam Help

https://powcoder.com

https://academichonesty.unimelb.edu.au/

Add WeChat powcoder

➢ All examinable work (GROK worksheet answers and all project work) that you submit for this subject must be your own!!

Academic Honesty

➢ Common causes of breaches in the past have been:

✓ friends asking to look over your code to ``get hints'' for their own project

✓ flatmates accessing your code via a shared desktop computer with saved login details

✓ study groups where the facilitator has overstepped the line and provided sample code to help people along

Academic Honesty

➢ Common attempts to escape undetected are:

  ✓ changing the comments but not the code

  ✓ changing variable names

  ✓ rearranging blocks of code (sometimes breaking the logic in the process!)

➢ It is all too easy to automatically pick up on all of these, and many, many more, approaches using software plagiarism detection software ... and we **do** check

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

So what is appropriate?

➤ You are encouraged to share/collaborate directly on code for any non-examinable items (notably the worksheets questions) … and you will learn a lot from sharing the code of others (including the sample solutions in the worksheets)

➤ You are very welcome to discuss with fellow classmates your *approach* to worksheet questions and the projects, in conceptual terms, or in terms of key data types or programming constructs used (just **not** with the aid of raw code)
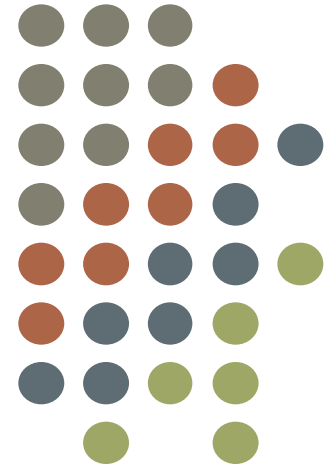
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Break!!!!

Lecture Objectives

After completing this lecture, you will be able to:

Assignment Project Exam Help

- Describe the basic features of an algorithm

https://powcoder.com

- Explain how hardware and software collaborate in a computer's architecture

Add WeChat powcoder

- Compose and run a simple Python program

Algorithms & Information Processing

➢ Computer science focuses on

Assignment Project Exam Help

- **Algorithms** https://powcoder.com
- **Information processing** Add WeChat powcoder

Algorithm – what is it?

➢ Sequence of steps that describes each of these computational processes is called an **algorithm**

➢ Features of an algorithm:

- Consists of a finite number of instructions

- Each individual instruction is well defined

- Describes a process that eventually halts after arriving at a solution to a problem

- Solves a general class of problems

Algorithm – example!

- Steps for making a cup of coffee !!!

➢ Imagine that you want a robot (or a friend) to help it.

➢ How will you explain this procedure to the robot (or your friend) so they can make it?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Algorithm – another example!

➢ Steps for subtracting two numbers:

- **Step 1:** Write down the numbers, with larger number above smaller one, digits columns aligned from right

- **Step 2:** Start with rightmost column of digits and work your way left through the various columns

- **Step 3:** Write down difference between the digits in the current column of digits, borrowing a 1 from the top number's next column to the left if necessary

- **Step 4:** If there is no next column to the left, stop

  - Otherwise, move to column to the left; go to Step 3

➢ The **computing agent** is a human being

Information Processing

➢ Information is also commonly referred to as **data**

➢ Data needs to be organized and stored in some form to be relevant and easily accessed!

➢ In carrying out the instructions of an algorithm, computing agent manipulates information

  ● Start with **input** → **do something/task** → produce **output**

Structure of a Modern Computer System

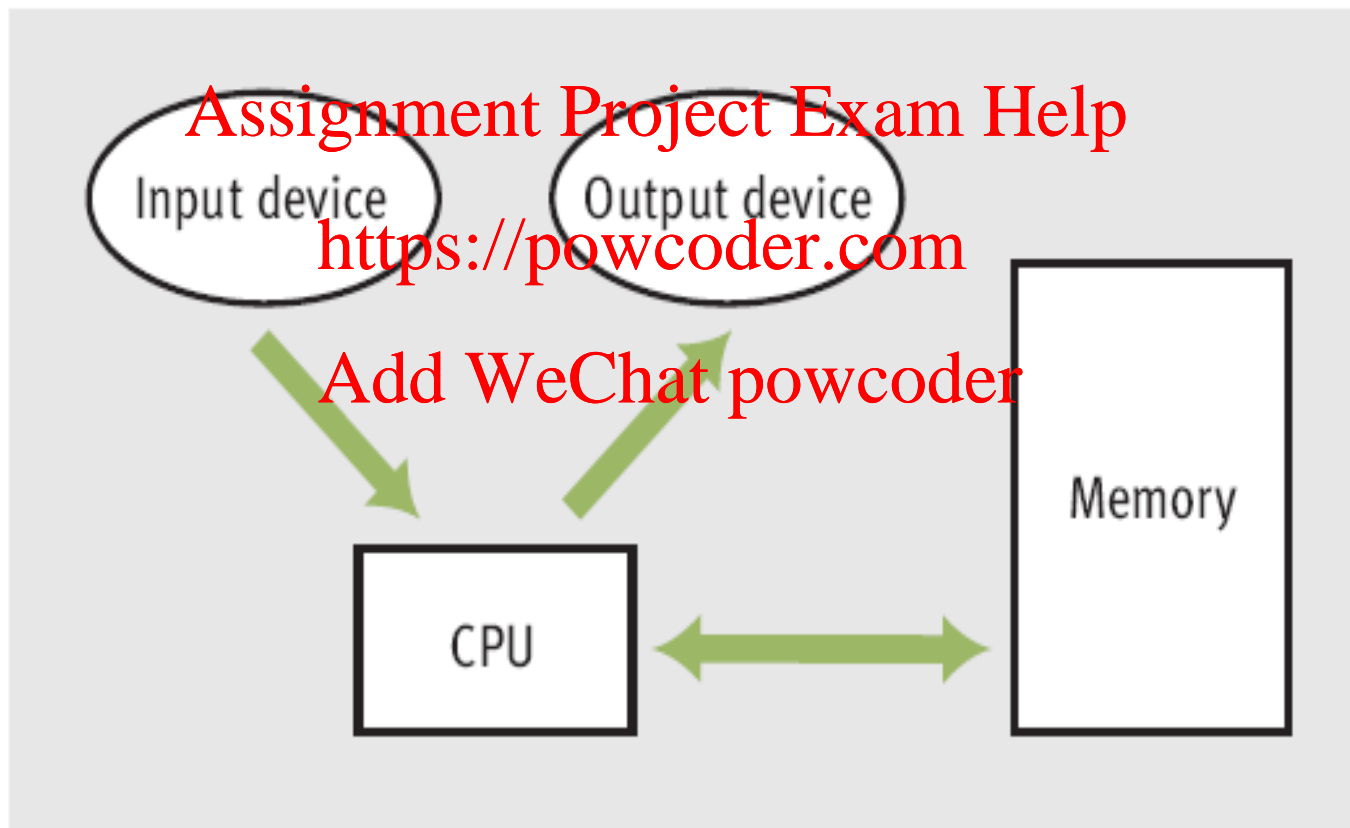➢ A modern computer system consists of **hardware** and **software**

- Hardware: physical devices required to execute algorithms

- Software: set of these algorithms, represented as **programs** in particular **programming languages**

Computer Hardware

Input device

Output device

CPU

Memory

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Computer Hardware (continued)

Cell 7	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1
Cell 6	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1
Cell 5												0	1	1	
Cell 4	1	0	1	1	0	1	1	1	1	1	0	1	1	1	
Cell 3	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1
Cell 2										0	1	1	1	0	1
Cell 1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1
Cell 0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0

➢ **Random access memory (RAM)** is also called **internal** or primary

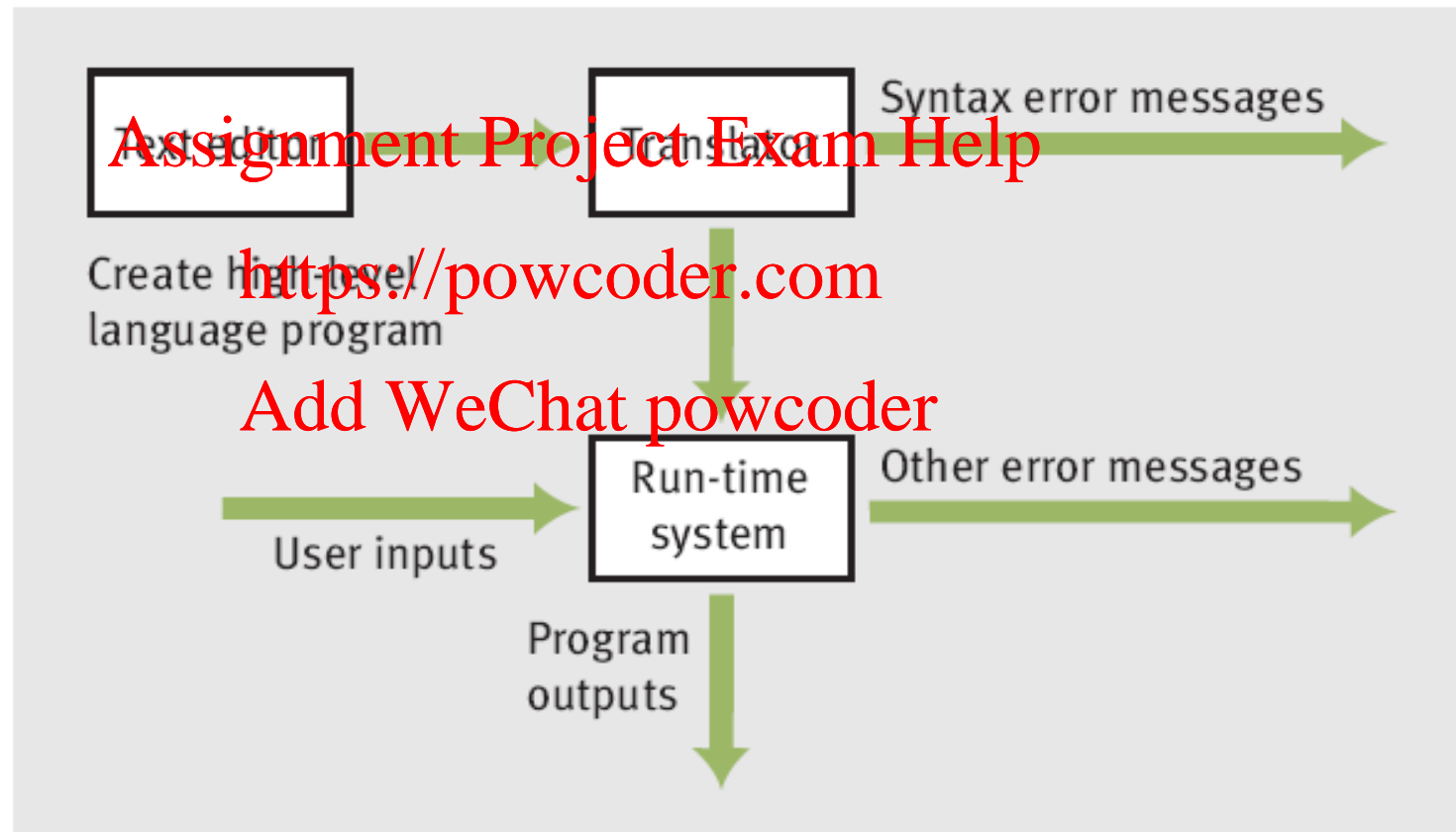➢ **External** or secondary memory example: cd; floppies magnetic disks; optical disks!!!

Computer Software

➢ A program stored in computer memory must be represented in binary digits, or **machine code**

➢ A **loader** takes a set of machine language instructions as input and loads them into the appropriate memory location

➢ The most important example of **system software** is a computer's **operating system**

  o Some important parts: **file system**, **user interfaces** (**terminal-based** or **GUIs**)

➢ **Applications** include Web browsers, games, etc.

Computer Software (continued)

Text editor   Translator   Syntax error messages

Create high-level
language program

Run-time
system   Other error messages

User inputs

Program
outputs

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Getting Started with Python Programming

- Early 1990s: Guido van Rossum
  - invented the Python programming language

- **Python** is a high-level, general-purpose programming language for solving problems on modern computer systems

- Useful resources at *www.python.org*

Running Code in the Interactive Shell

➢ Python is an **interpreted** language

➢ Simple Python expressions and statements can be run in the **shell**

- Easiest way to open a Python shell is to launch the IDLE (Integrated **D**eve**L**opment **E**nviron**m**ent)

- To quit, select the window's close box

- Shell or command line is useful for:

  - Experimenting with short expressions or statements

  - Consulting the documentation

Running Code in the Interactive Shell (continued)

```
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 4 | Col: 4

Input, Print, String Processing & Output

➢ Programs usually accept inputs from a source, process them, and output results to a destination

- In terminal-based interactive programs, these are the keyboard and terminal display

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

The **print** statement

A **print** statement evaluates the expression and displays them, separated by one space, in the console window

- **Syntax:**

print (<expression>, …, <expression>)

Example:

>>>print (3)

3

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- If you want to display a text (in python called a string), then the string must be enclosed within quotations – single or double

Strings in python

- The word **string** is used to describe a piece of text inside quotes.
- You've seen how Python can be used to manipulate numbers, but it is also useful to be able to manipulate text.
- The things you type into a Python program are expected to be Python commands.

Assignment Project Exam Help

https://powcoder.com

Example:

>>>print ("hi there")

Add WeChat powcoder

>>>print ('hi there')

>>>print ("what's she doing")

Lets run this and see what happens!!!!!

Strings in python

- Suppose you want to represent the words **Hello There** as text inside a Python program. If you type them as is into the program directly, Python will try to treat those words as names in the Python language:

```
>>> print (hello there)
```

What happens? Lets check it out!

We get an error message!!!!!

Errors and **python translator**

- Sometimes, you will write something which the Python interpreter does not understand. The most basic kind of mistake you can make is a **syntax error**.

- A syntax error occurs when you type something which is not properly formed, according to the rules of the programming language.

- It is the same as a grammatical error in English (or any natural language for that matter).

- So, in this example it did not know what hello there is (not an identifier) until you tell it to be a string within quotes.

  >>> print ('hello there')

  hello there

print - examples

```
print(<expression>)
```

```
>>> print('Hi there')
Hi there
```

```
print(<expression>, ... , <expression>)
```

- You can type in expressions and it will calculate the answer for you (noting that the asterisk, or *, signifies multiplication):

```
>>>print (3 * 2)
6
```

The **input** statement

- Python has a built-in function called **input** that can be used to get keyboard input from the user as a string.

  Assignment Project Exam Help

- The **input** function can be given a message to display, usually prompting the user with what kind of information the program wants.
  https://powcoder.com

  Add WeChat powcoder

- A **variable** is a name for a value. When a variable receives a value, that value can be accessed using the variable name when required.

The **input** statement - example

```
>>> name = input("Enter your name: ")
Enter your name: Ken Lambert
>>> name
'Ken Lambert'
>>> print(name)
Ken Lambert
>>>
```

Assignment Project Exam Help

https://powcoder.com

```
<variable identifier> = input(<a string prompt>)
```

Add WeChat powcoder

```
>>> name
'Ken Lambert'
```

```
>>> first = int(input("Enter the first number: "))
Enter the first number: 23
>>> second = int(input("Enter the second number: "))
Enter the second number: 44
>>> print("The sum is", first + second)
The sum is 67
>>>
```

Variables and the Assignment Statement

- A **variable** associates a name with a value
  - Makes it easy to remember and use later in program
- Variable naming rules:
  - Reserved words cannot be used as variable names
    - Examples: **if**, **for**, **while** and **import**
  - Name must begin with a letter or underscore _
- The rest of the name can contain zero or more occurrences of the following things:
  - Digits (**0** to **9**) .
  - Alphabetic letters.
  - Underscores.
- Names are case sensitive
  - Example: **WEIGHT** is different from **weight**

Tip: use "camel casing" (Example: **interestRate**)

Variables and the Assignment Statement (continued)

- Nothing else is allowed in a variable name. Here is (incomplete) list of the things you can't use anywhere in a variable name:
  - Whitespace characters (the space, tab, new line).
  - Hyphens (-)
  - Quotation marks (single or double)
  - Symbol characters such as: the question mark, the exclamation mark, brackets, and so forth.
- The following words having special meaning in Python (reserve words), and so they cannot be used for variable

  Examples:

  def, if, for, while, else, except, return, True, range, list, continue

Variables and the Assignment Statement (continued)

- Programmers use all uppercase letters for **symbolic constants**
  - ➢ Examples: **TAX_RATE** and **STANDARD_DEDUCTION**

- Variables receive initial values and can be reset to new values with an **assignment statement**

<variable name> = <expression>

  - ➢ Subsequent uses of the variable name in expressions are known as **variable references**

Variables : Example 1

#Which of the following are valid variable names?

a. length
b. _width
c. firstWord
d. 2MoreToGo
e. halt!

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Solution: ????

Editing, Saving, and Running a Script

- We can then run Python program files or **scripts** within IDLE or from the OS's command prompt

- Python program files use .py extension

- Running a script from IDLE allows you to construct some complex programs, test them, and save them in **program libraries** to reuse or share with others

Editing, Saving, and Running a Script

➢ Select New File from the File menu

➢ enter the python code/statements

➢ File/Save as (give a sensible name to your file. The extension will be.py)

➢ To run the file or code as a python script, select Run module from the Run Menu

Examples using IDLE – shown in lecture

Editing, Saving, and Running a Script (continued)

```
width = int(input("Enter the width: "))
height = int(input("Enter the height: "))
area = width * height
print("The area is", area, "square units")
```

Ln: 4 Col: 26

Editing, Saving, and Running a Script (continued)

```
>>> ================================ RESTART =========================
==========
>>>
Enter the width: 33
Enter the height: 22
The area is 726 square units.
>>>
```

Ln: 9 Col: 4

Behind the Scenes: How Python Works

Python code → Syntax Checker and Translator → Syntax error messages

Byte code

User inputs → Python Virtual Machine (PVM) → Other error messages

Program outputs

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder
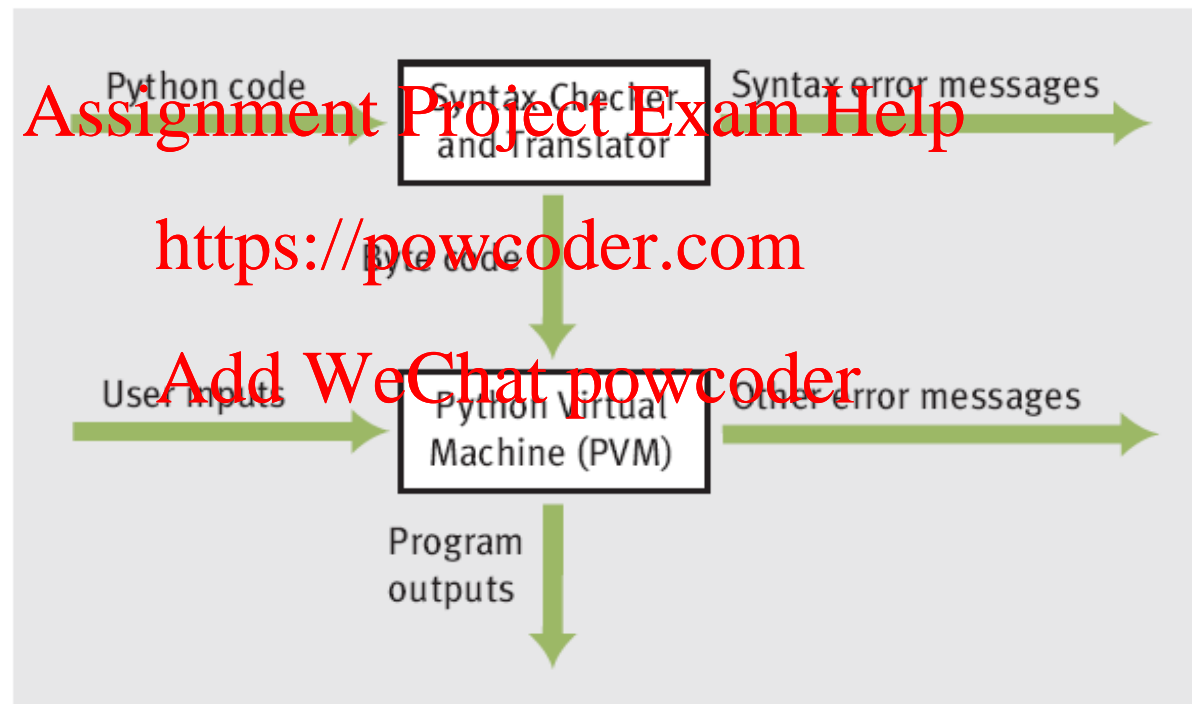
Detecting and Correcting Syntax Errors

➢ Programmers inevitably make typographical errors when editing programs; called **syntax errors**

- The Python interpreter will usually detect these

➢ **Syntax:** rules for forming sentences in a language

➢ When Python encounters a syntax error in a program, it halts execution with an error message

Detecting and Correcting Syntax Errors (continued)

```
>>> length = int(input("Enter the length: "))
Enter the length: 44

>>> print(length)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
NameError: name 'length' is not defined
```

```
>>>   print length
  File "<pyshell#1>", line 1
    print length
          ^
SyntaxError: unexpected indent
```

```
>>> 3 +
    3 +
SyntaxError: invalid syntax
```

Install Python

- We will use Python v3.4 via GROK (on my machine 3.5.1)
- You just write it like a text and the python interpreter turns it into machine code for you
- Get a copy of python for your own machine at home - there are free versions for Windows, MacOS and Linux
  http://www.python.org/download/
- Portable version (USB) http://portablepython.com/
- Advanced Python Distribution (for scientific experimentation)
  http://www.enthought.com/products/edudownload.php

Grok Learning environment

- GROK Learning is the web-based programming environment we will be using for the duration of this subject in your labs:

  https://groklearning.com/course/unimelb-isys90088-2018-s2/

- All you need to access the system is a browser, an internet connection and your GROK account

- Different modes of working in GROK: code, run, mark, terminal

Summary

- ➢ Fundamental ideas of computer science
  - The algorithm
  - Information processing
- ➢ Real computing agents can be constructed out of hardware devices
  - CPU, memory, and input and output devices

Summary (continued)

➢ Software provides the means whereby different algorithms can be run on a general-purpose hardware device

- Written in programming languages

➢ Languages such as Python are high-level

➢ Interpreter translates a Python program to a lower-level form that can be executed on a real computer

➢ Python shell provides a command prompt for evaluating and viewing the results of Python expressions and statements

Summary (continued)

➢ IDLE is an integrated development environment that allows the programmer to save programs in files and load them into a shell for testing

➢ Python scripts are programs that are saved in files and run from a terminal command prompt

➢ When a Python program is executed, it is translated into byte code
- Sent to PVM for further interpretation and execution

➢ Syntax: set of rules for forming correct expressions and statements in a programming language