MODULE - I (PROGRAMMING ENVIRONMENT AND PYTHON BASICS)

MODULE 1

- Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works.
- The software development process –A case study.
- Basic coding skills – strings, assignment, and comments, Numeric data types and character sets, Expressions, Using inbuilt functions and modules.
- Control statements – Iteration with for/while loop, Formatting text for output, A case study, Selection structure (if-else, switch- case), Conditional iteration with while, A case study, Testing control statements, Lazy evaluation.

INTRODUCTION TO PYTHON

- high-level, interpreted, interactive and object-oriented scripting language.
- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

GETTING STARTED WITH PYTHON PROGRAMMING

- You can download Python, its documentation from
- http://www.python.org/download/
- To launch an interactive session with Python's shell from a terminal
- command prompt, open a terminal window, and enter python or python3 at the prompt.

INTERACTIVE SHELL,

- Simple Python expressions and statements in an interactive programming environment called the ***shell***.
- A shell window contains an opening message followed by the special symbol >>>, called a shell prompt.
- When you enter an expression or statement, Python evaluates it and displays its result

IDLE

- The easiest way to open a Python shell is to launch the IDLE\
- This is an Integrated program Development Environment
- The IDLE programming environment uses color-coding to help the reader pick out different elements in the code.
- For example, the items within quotation marks are in green, the names of standard functions are in purple, program comments are in red, and the responses of IDLE to user commands are in blue.

IDLE – WORKOUTS……

```
>>> 3 + 4                          # Simple arithmetic
7
>>> 3                              # The value of 3 is
3
>>> "Python is really cool!"       # Use a string for text
'Python is really cool!'
>>> name = "Ken Lambert"           # Give a variable a value
>>> name                           # The value of name is
'Ken Lambert'
>>> "Hi there, " + name            # Create some new text
'Hi there, Ken Lambert'
>>> print('Hi there')              # Output some text
Hi there
>>> print("Hi there,", name)       # Output two values
Hi there, Ken Lambert
```

Color	Type of Element	Examples
Black	Inputs in the IDLE shell Numbers Operator symbols Variable, function, and method references Punctuation marks	`67, +, name, y = factorial(x)`
Blue	Outputs in the IDLE shell Function, class, and method names in definitions	`'Ken Lambert'`, `def factorial(n)`
Green	Strings	`"Ken Lambert"`
Orange	Keywords	`def, if, while`
Purple	Built-in function names	`abs, round, int`
Red	Program comments Error messages in the IDLE shell	`# Output the results` `ZeroDivisionError: division by zero`

DETECTING AND CORRECTING SYNTAX ERRORS

- Python interpreter will nearly always detect errors.
- Such errors are called syntax errors.
- The term *syntax* refers to the rules for forming sentences in a language.
- When Python encounters a syntax error in a program, it halts execution with an error message.
- Python shell show several types of syntax errors and the corresponding error messages:

SAMPLE ERRORS

```
>>> length = int(input("Enter the length: "))
Enter the length: 44
>>> print(lenth)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
NameError: name 'lenth' is not defined
```

```
>>> print(length)
SyntaxError: unexpected indent
```

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

WORK……

1. Open a Python shell, enter the following expressions, and observe the results:

   a. **8**
   b. **8 * 2**
   c. **8 ** 2**
   d. **8/12**
   e. **8 // 12**
   f. **8/0**

PYTHON STRING

- Strings are sequences of characters.
- To create a string, put the sequence of characters inside either single quotes, double quotes, or triple quotes and then assign it to a variable.

```
>>> single_quote_character = 'a'
>>> print(single_quote_character)
a
>>> print(type(single_quote_character)) # check the type of the variable.
<class 'str'>
```

ASSIGNMENT STATEMENTS IN PYTHON

- **Python assignment statements** is used to assign objects to names.
- The target of an assignment statement is written on the left side of the equal sign (=), and the object on the right.

ASSIGNMENT STATEMENTS IN PYTHON

- **Basic form:** most common form

```
student = 'abc'
print(student)
```

- **Tuple assignment:**

```
# equivalent to: (x, y) = (50, 100)
x, y = 50, 100

print('x = ', x)
print('y = ', y)
```

- **List assignment:** This works in the same way as the tuple assignment.

```
[x, y] = [2, 4]

print('x = ', x)
print('y = ', y)
```

- **Sequence assignment:** Python assigns the items one at a time by position.

```
a, b, c = 'HEY'

print('a = ', a)
print('b = ', b)
print('c = ', c)
```

INPUT, PROCESSING, AND OUTPUT

- In terminal-based interactive programs, the input source is the keyboard, and the output destination is the terminal display.

  Inputs : Python expressions or statements.

  outputs : Results displayed in the shell.

- output functions

  print(<expression>) - Python first evaluates the expression and then displays its value.

  >>> print('Hi there')

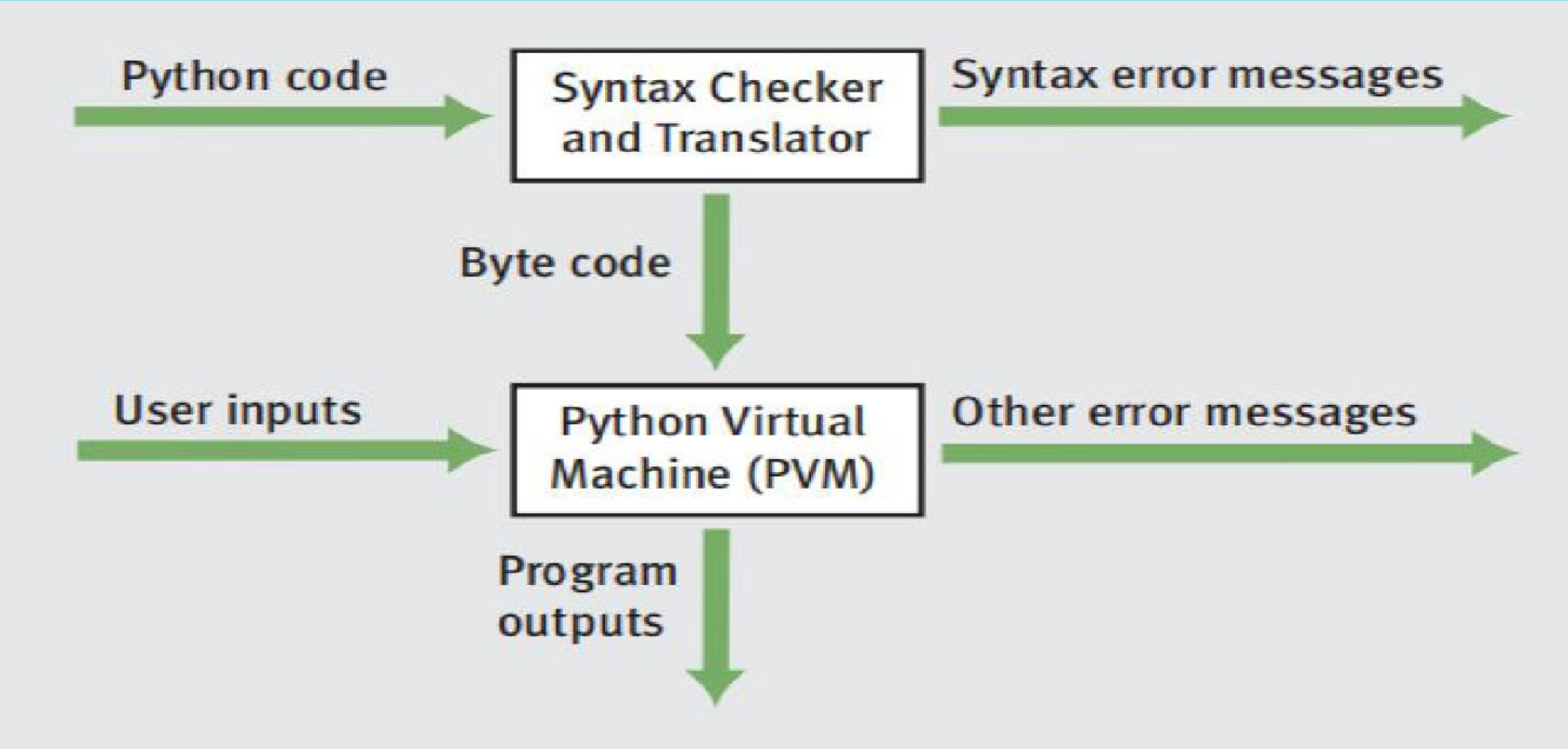
  Hi there

INPUT FUNCTIONS

input() - This function causes the program to stop and wait for the user to enter a value from the keyboard.

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

BEHIND THE SCENES: HOW PYTHON WORKS



Steps in interpreting a Python program

1) The interpreter reads a Python expression or statement, also called the source code, and verifies that it is well formed.

2) If a Python expression is well formed, the interpreter then translates it to an equivalent form in a low-level language called byte code.

3) This byte code is next sent to another software component, called the Python virtual machine (PVM), where it is executed. If another error occurs during this step, execution also halts with an error message.

THE SOFTWARE DEVELOPMENT PROCESS

Waterfall Model - Different Phases

1. Customer request—In this phase, the programmers receive a broad statement of a problem that is potentially amenable to a computerized solution. This step is also called the user requirements phase.

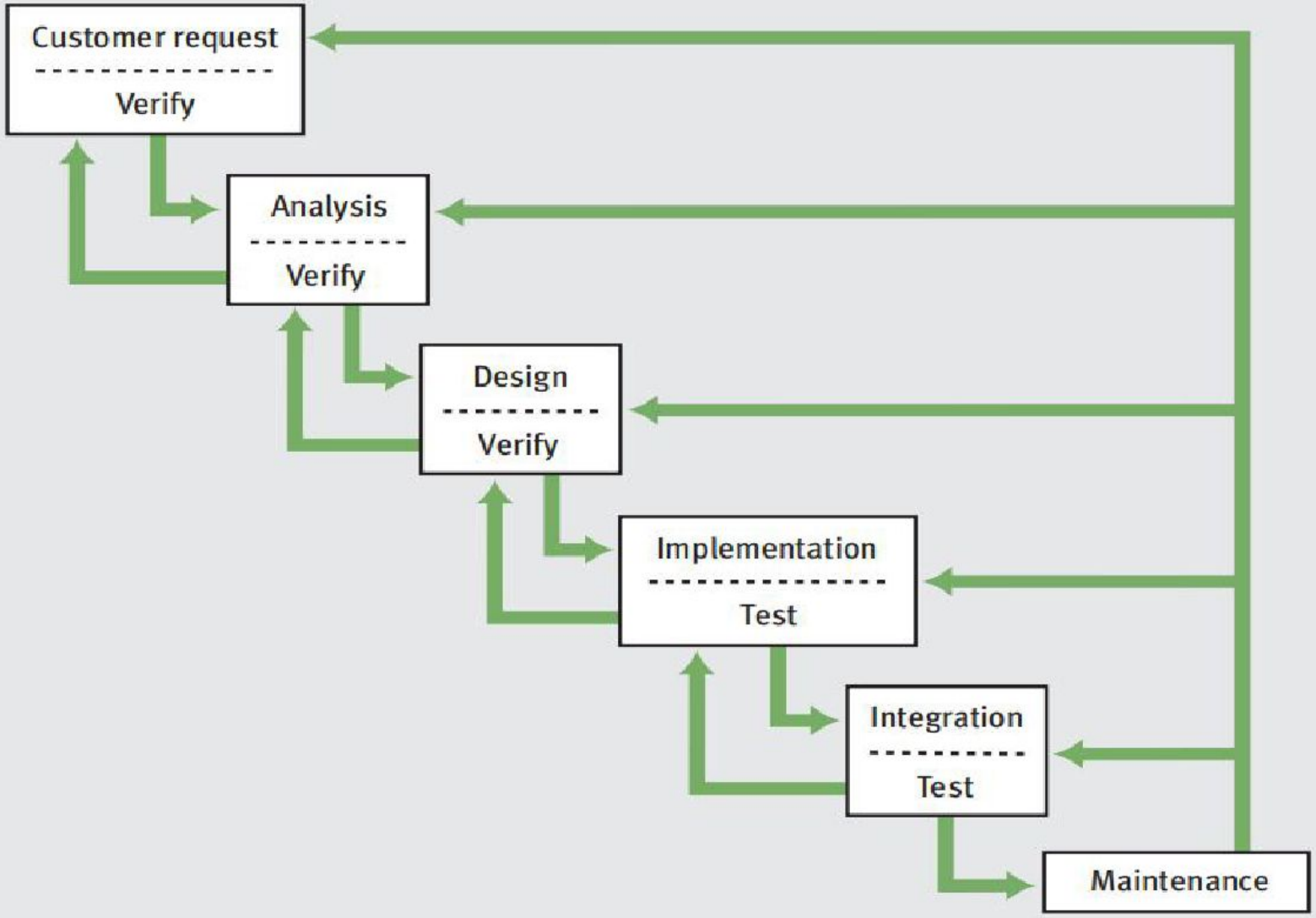
2. Analysis—The programmers determine what the program will do. This is sometimes viewed as a process of clarifying the specifications for the problem.

3. Design—The programmers determine how the program will do its task.

4) Implementation—The programmers write the program. This step is also called the coding phase.

5) Integration—Large programs have many parts. In the integration phase, these parts are brought together into a smoothly functioning whole, usually not an easy task.

6) Maintenance—Programs usually have a long life; a lifespan of 5 to15 years is common for software. During this time, requirements change, errors are detected, and minor or major modifications are made


Customer request
Verify
Analysis
Verify
Design
Verify
Implementation
Test
Integration
Test
Maintenance

CASE STUDY : INCOME TAX CALCULATOR

Request:

The customer requests a program that computes a person's income tax.

Analysis:

Analysis often requires the programmer to learn some things about the problem domain, in this case, the relevant tax law.

- All taxpayers are charged a flat tax rate of 20%.
- All taxpayers are allowed a $10,000 standard deduction.
- For each dependent, a taxpayer is allowed an additional $3,000 deduction.
- Gross income must be entered to the nearest penny.
- The income tax is expressed as a decimal number.

Another part of analysis determines what information the user will have to provide.

user inputs - gross income and number of dependents.

Design:

we describe how the program is going to do it.

This usually involves writing an algorithm - pseudocode

1) Input the gross income and number of dependents

2) Compute the taxable income using the formula

*3) Taxable income = gross income - 10000 - (3000 * number of dependents)*

4) Compute the income tax using the formula

*5) Tax = taxable income * 0.20*

6) Print the tax

Implementation (Coding):

Given the preceding pseudocode, an experienced programmer would now find it easy to write the corresponding Python program.

Testing:

Only thorough testing can build confidence that a program is working correctly.

Testing is a deliberate process that requires some planning and discipline on the programmer's part.

A correct program produces the expected output for any legitimate input.

Testing all of the possible combinations of inputs would be impractical.

The challenge is to find a smaller set of inputs, called a test suite, from which we can conclude that the program will likely be correct for all inputs.

```
# Initialize the constants
TAX_RATE = 0.20
STANDARD_DEDUCTION = 10000.0
DEPENDENT_DEDUCTION = 3000.0
# Request the inputs
grossIncome = float(input("Enter the gross income: "))
numDependents = int(input("Enter the number of dependents: "))
# Compute the income tax
taxableIncome = grossIncome - STANDARD_DEDUCTION - \
DEPENDENT_DEDUCTION * numDependents
incomeTax = taxableIncome * TAX_RATE
# Display the income tax
print("The income tax is $" + str(incomeTax))
```

Number of Dependents	Gross Income	Expected Tax
0	10000	0
1	10000	–600
2	10000	–1200
0	20000	2000
1	20000	1400
2	20000	800

Table 2-1 The test suite for the tax calculator program

CASE STUDY :EXERCISE

Five Star Video rents VHS tapes and DVDs to the same users who like to buy LP record albums. The store rents new videos for $3.00 a night, and
oldies for $2.00 a night. Write a program that the clerks at Five Star Video
can use to calculate the total charge for a customer's video rentals. The program should prompt the user for the number of each type of video and output the total
cost.

BASIC CODING SKILLS : STRINGS

STRINGS

- String Literals: In Python, a string literal is a sequence of characters enclosed in single or double quotation
- marks.