PHINMA EDUCATION
MAKING LIVES BETTER THROUGH EDUCATION

ITE 306: Integrative Programming and Technologies
Module #1

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

Lesson title: ORIENTATION DAY for ITE 306
Lesson Objectives:
At the end of this module, I should be able to:
1. Overview of the course outline
2. Know what is Integrative Programming

Materials:
Activity Sheets, smart phone or computer, internet connection
References:
- https://muele.muni.ac.ug/course/info.php?id=68
- https://pediaa.com/what-is-the-difference-between-interpreted-and-compiled-language/

Productivity Tip:
"Set attainable goals. An excellent way to increase productivity is by setting attainable goals. The reason for this is that seeing future goals being reached serves as motivation to continue working."

A. LESSON PREVIEW/REVIEW

1) Introduction (2 mins)

What is integrative programming and technologies?

This course focuses on the use of several technology components to communicate and work together with each other. It covers numerous topics related to inter-system communication, integrative coding, scripting techniques, IoT, web services and an overview of programming language.

Course Outline and Schedule		
Day	**Sched.**	**Topics**
1	IN	ORIENTATION DAY for ITE 306 - Introduction to Integrative Programming
2	IN	Python Basics
3	OUT	The Fundamentals of Python Programming Language
4	OUT	Python Variables
5	IN	Python Datatypes
6	IN	FIRST PERIOD QUIZ
7	OUT	Python Operators and Input Functions
8	OUT	Python Lists and Tuples
9	IN	FIRST PERIOD EXAM

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

10	**IN**	**Python Sets**
11	**OUT**	**Python Dictionaries**
12	**OUT**	**Conditions and Statements: Python If….else**
13	**IN**	**SECOND PERIOD QUIZ**
14	**IN**	**Python Loops**
15	**OUT**	**Python Functions**
16	**OUT**	**Python Modules**
17	**IN**	**SECOND PERIOD EXAM**
18	**IN**	**Web Service Standards and Architecture: SOAP and REST**
19	**OUT**	**REST API using Flask: Building a webpage using Python**
20	**OUT**	**Flask Templates and Static Files**
21	**IN**	**THIRD PERIOD QUIZ**
22	**IN**	**Internet of Things (IoT)**
23	**OUT**	**IoT with Python: Essential Package (mysqldb)**
24	**OUT**	**Cloud Computing**
25	**IN**	**Cloud Computing: Azure vs. AWS vs. Google Cloud**
26	**IN**	**THIRD PERIOD EXAM**

2) Activity 1: What I Know Chart, part 1 (3 mins)

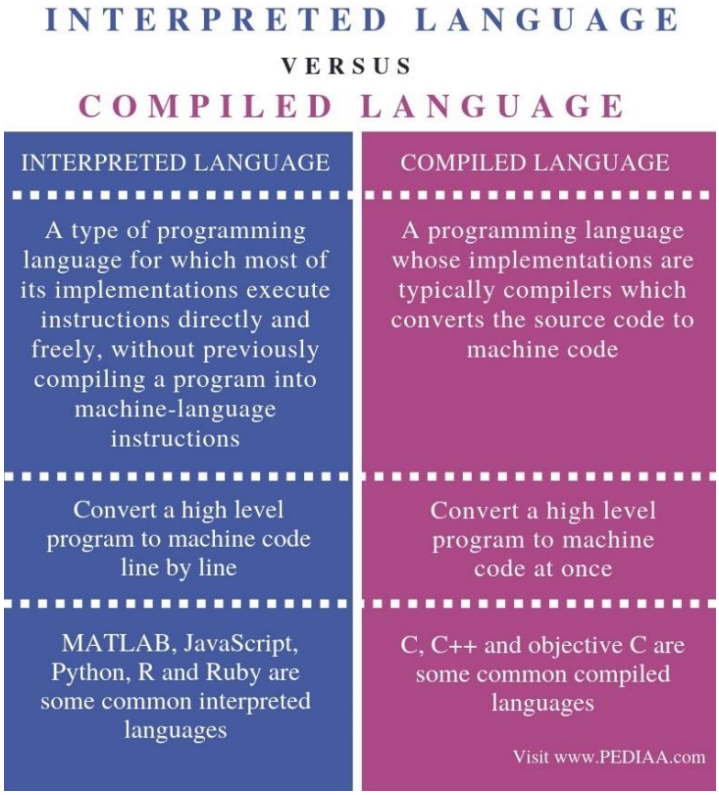
Try answering the questions below by writing your ideas under the first column *What I Know*. It's okay if you write key words or phrases that you think are related to the questions. Leave the third column blank for Activity 4.

What I Know	Questions:	What I Learned (Activity 4)
	1. What are the programming languages that you already know?	
	2. What are the software compilers that you already used?	

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

B. MAIN LESSON

3) Activity 2: Content Notes (13 mins)



What is Interpreted Language?

An interpreted language is a language that is based on an interpreter. Wherein, an **interpreter** is a software that is capable of converting a high-level program into machine-understandable machine code. The interpreter executes the program directly and translates each statement into machine code in a sequential manner. In other words, an interpreter converts the source code into machine code line after the line. Here, the instruction set is a **bytecode**.

Moreover, an interpreter displays an error at a time. Therefore, the programmer should fix that error to interpret the next line. Some examples of interpreted languages are MATLAB, JavaScript, Python, R and Ruby. Furthermore, some languages are implemented using both compiler and interpreter. For example, C# and Java compile into bytecode, which is a virtual machine friendly, interpreted language.

What is Compiled Language?

A language that depends on the compiler is a compiled language. Herein, a compiler is a software that converts the source code to machine code at once. If there are syntactic or semantic errors, the compiler will indicate them. However, the compiler checks the whole program and displays all errors on the console. It is not possible to execute the program without fixing the errors.

Usually, the execution time of compiled languages is lower. In other words, these languages execute faster. Therefore, these programming languages help to develop real-time embedded systems, games, operating systems, database applications which require fast processing. For example, some common compiled languages are C and C++.

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

What is the Difference Between Interpreted and Compiled Language?

- **interpreted language converts the source code into machine code line by line**
- **compiled language converts the source code into machine code at once.**

A computer program is a set of instructions that instruct the CPU to perform the defined task or tasks. There are various programming languages.

- **High-level programming languages** have a syntax similar to the English language. Therefore, it is easier for the programmer to read and understand these programs. On the other hand, the machine does not understand these programs. Therefore, it is necessary to convert these programs to machine understandable **machine code**.

- **Low-level languages** are one step below high-level programming languages. These languages are closer to the hardware level than high-level languages. Thus, it is necessary to convert whatever the language the programmer uses to write the program into machine code for the CPU to perform the task.

Before we begin, fill-up the information sheet by providing your personal information.

Cut above this line and give this form to your teacher for record keeping

STUDENT INFORMATION SHEET	
NAME:	
DATE OF BIRTH:	
E-MAIL ADDRESS:	
CONTACT NUMBER:	
COMPLETE ADDRESS:	
PARENT'S / GUARDIAN NAME:	
SUBJECT CODE AND NAME:	
SECTION AND TIME:	

Let's create a Group chat and name it as: ***ITE306 - <Section Name>***

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

4) Activity 3: Skill-building Activities
Answer the following questions based on your personal experience.

1.) What is your programming experience?

_____
_____
_____
_____
_____

2.) What is your main computer programming languages?

_____
_____
_____
_____

3.) How do you describe programming?

_____
_____
_____
_____
_____

4.) How difficult is coding for you?

_____
_____
_____
_____
_____

5.) What's the hardest coding language for you?

_____
_____
_____
_____
_____

5) Activity 4: What I Know Chart, part 2
Review the questions in the *What I Know Chart* from Activity 1 and write your answers to the questions based on what you **now** know in the third column of the chart.

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

6) Activity 5: Check for Understanding
Identify what is being describe, write your answer on the space provided before each number.

_____1. These languages are closer to the hardware level than high-level languages.
_____2. These programming languages have a syntax similar to the English language.
_____3. It is a software that converts the source code to machine code at once.
_____4. It is a software that converts the source code into machine code line after the line.
_____5. It is a language that depends on the compiler.
_____6. It is a language that is based on an interpreter.
_____7. Javascript, Python, Ruby are some common _____ language.
_____8. C and C++ are some common example of _____ language.

C. LESSON WRAP-UP

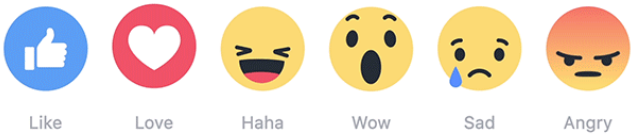
Activity 6: Thinking about Learning

Congratulations for finishing this module! **Shade** the number of the module that you finished to track how much work you have accomplished and how much work there is left to do.

You are done with the session! Let's track your progress.

Period 1											Period 2											Period 3								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Rate the session for today by encircling the emoji that best captures your experience and write your reason for choosing that emoji.



Reason/s: _____

FAQs

1. **What is the programming language that we will use in Integrative Programming and Technologies (ITE-036) subject?**
   Answer: *We will be using one of the easiest programming languages to learn the Python. One of the design principles behind Python is a commitment to making the act of programming as enjoyable as possible.*

2. **Is Python Free?**
   Answer: *Yes, Python is completely free and open source.*
   One of the best aspects of Python is its community-driven development, which is mainly done through GitHub (for checking out the language's source code and submitting patches) and IRC (where users can discuss bugs, features, and other Python-related topics).

Name: _____ Class number: _____
Section: _____ Schedule: _____ Date: _____

KEY TO CORRECTIONS
Answers to Checking for Understanding:

1. ***Low-level languages*** These languages are closer to the hardware level than high-level languages.
2. ***High-level languages*** These programming languages have a syntax similar to the English language.
3. ***Compiler*** It is a software that converts the source code to machine code at once.
4. ***Interpreter*** It is a software that converts the source code into machine code line after the line.
5. ***Compiled languages*** It is a language that depends on the compiler.
6. ***Interpreted languages*** It is a language that is based on an interpreter.
7. ***Interpreted*** Javascript, Python, Ruby are some common _____ language.
8. ***Compiled*** C and C++ are some common example of _____ language.