

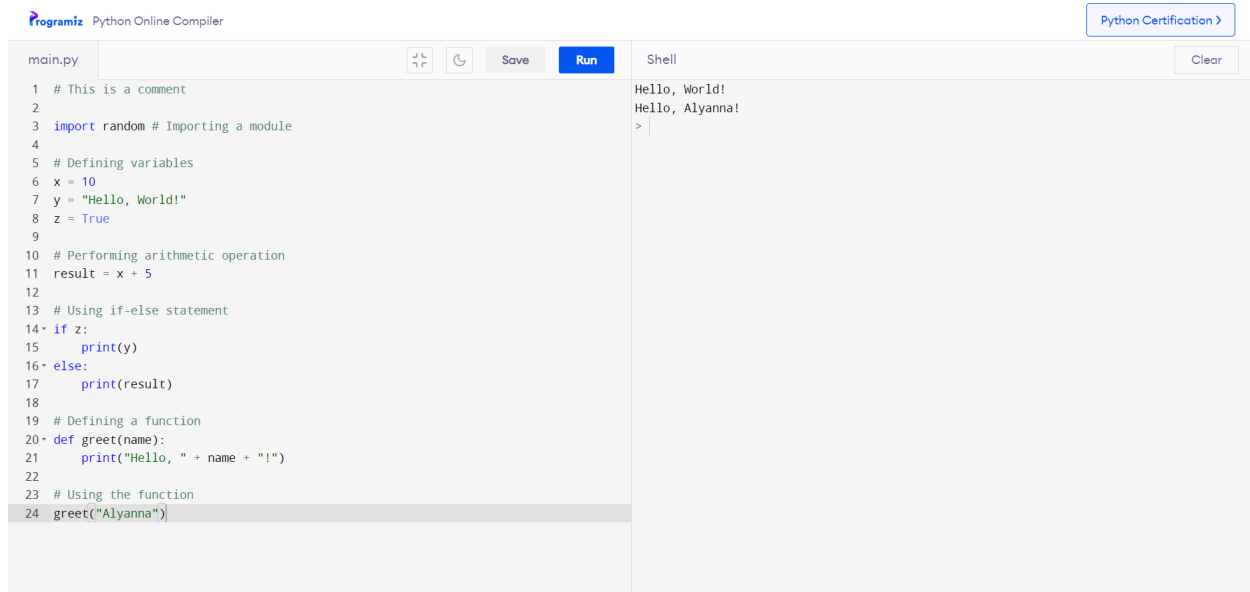
NAME: Alyanna Marie N. Bilazon
SECTION: CS3C

Introduction to python

- **Features of Python Language**
 - **Dynamic Typing:** Because Python is dynamically typed, variables can have values assigned to them without their data types being to be explicitly declared.
 - **Interpreted Language:** Python is an interpreted language, meaning that debugging and testing it is simple because the Python interpreter runs the source code line by line.
 - **High-level Language:** Python is a high-level, simple-to-learn programming language that abstracts away low-level features.
 - **Object-Oriented:** Python enables programmers to write reusable and modular code by supporting the concepts of object-oriented programming.
 - **Extensive Standard Library:** There is less need for external dependencies with Python thanks to its extensive standard library, which contains modules and packages for a variety of applications.
 - **Dynamically Typed:** Python is dynamically typed, which means that variables are interpreted at runtime and you don't have to define their data type when you declare them.
- **Advantages of Python**
 - **Ease of Learning:** Novice programmers will find Python's syntax easily comprehensible and easy to learn.
 - **Community and Documentation:** There is a large and vibrant Python community that offers a wealth of guides, tutorials, and assistance.
 - **Python's versatility** makes it appropriate for a wide range of applications, such as automation, data science, web development, artificial intelligence, and machine learning.
 - **Quick Development:** Python allows for quicker development and experimentation because of its high-level abstractions and simplicity.
 - **Extensive Libraries:** Developers can complete jobs quickly and effectively because to the extensive third-party libraries and the robust standard library.
 - **Platform Independence:** Python programs are cross-platform executables.
- **Disadvantages of Python Programming Language**
 - **Execution Speed:** Python is an interpreted language, which means that it may execute more slowly than compiled languages like C or C++.
 - **Global Interpreter Lock (GIL):** By permitting only one thread to run Python bytecode at a time, the Global Interpreter Lock can reduce the efficiency of multithreaded Python systems.
 - **Mobile Development:** Other languages, such as Swift (for iOS) and Kotlin (for Android), are more frequently utilized for developing mobile applications than Python.
 - **Memory Consumption:** Python might use more memory than programming languages like C or C++, which could be an issue in settings with limited resources.

- Applications of Python Programming Language
 - Web development: Python is a popular choice for creating dynamic and scalable web applications because of frameworks like Django and Flask.
 - Data Science and Machine Learning: Because to libraries like NumPy, Pandas, and TensorFlow, Python is extensively used in data analysis, machine learning, and artificial intelligence.
 - Automation and Scripting: Python's ease of use makes it a good choice for system management, task scheduling, and automation scripts.
 - Game creation: Pygame and other libraries offer a platform for making basic games, and Python is a useful language for game creation.
 - Desktop GUI programs: Python may be used to create GUI-heavy desktop programs by utilizing Tkinter and other tools.
 - Network Servers and Protocols: Several networking protocols can be implemented and network servers can be created using Python.
- What is Python?
 - Python is a general-purpose, interpreted, high-level programming language. Guido van Rossum was the creator, and it was originally published in 1991. Python is a great language for both novice and seasoned developers because it places a strong emphasis on readability, simplicity, and ease of use.
- Why Python?
 - Readability: Compared to other programming languages, Python's syntax is made to be easily understood and legible, enabling developers to convey ideas in fewer lines of code.
 - Versatility: Python is adaptable to a wide range of application areas since it supports several programming paradigms, such as procedural, object-oriented, and functional programming.
 - Extensive Library: Python offers modules and tools for a variety of jobs thanks to its extensive ecosystem of third-party libraries and frameworks, in addition to its rich standard library.
 - Community Support: The development and upkeep of Python are aided by a sizable and vibrant community. As a result, there is a lot of community assistance, tutorials, and documentation.
 - Platform Independence: Python code can function without modification on a variety of operating systems since it is platform-independent.
 - Integration Capabilities: Python's usefulness may be expanded and interoperability made easier by its ease of integration with other languages and technologies.
 - Rapid Prototyping: Python is a great option for quick prototyping and development because of its expressiveness and simplicity.
- How to install Python?
 - Go to the "Downloads" section of python.org to access the official Python website.
 - Download Python: Select the installer for the version of Python that works best for your operating system (Windows, macOS, or Linux).
 - Launch the Installer: Launch the installer that you downloaded, and make sure to select the option to add Python to the system PATH when the installation is taking place.
 - Check Installation: To make sure Python is installed correctly, open a terminal or command prompt and execute `python -V` or `python --version`.

- How to install PyCharm (IDE)?
 - Visit the JetBrains Website: Go to the PyCharm IDE part of the JetBrains website.
 - Download PyCharm: Download PyCharm in the Community or Professional edition that corresponds to your operating system.
 - Install PyCharm: To install PyCharm on your computer, launch the installer that you downloaded and adhere to the on-screen directions.
 - Set Up PyCharm: After installing PyCharm, run it. You might need to adjust the interpreter settings so that it uses the Python version that is installed on your computer.
- Basic Python Program Structure



The screenshot shows the Programiz Python Online Compiler interface. The top bar includes the Programiz logo, the text "Python Online Compiler", and a "Python Certification >" button. Below the bar, there's a tab labeled "main.py" and a toolbar with icons for undo, redo, save, and run. The main editor area contains the following Python code:

```
1 # This is a comment
2
3 import random # Importing a module
4
5 # Defining variables
6 x = 10
7 y = "Hello, World!"
8 z = True
9
10 # Performing arithmetic operation
11 result = x + 5
12
13 # Using if-else statement
14 if z:
15     print(y)
16 else:
17     print(result)
18
19 # Defining a function
20 def greet(name):
21     print("Hello, " + name + "!")
22
23 # Using the function
24 greet("Alyanna")
```

To the right of the code editor is a "Shell" window showing the output of the program:

```
Hello, World!
Hello, Alyanna!
>
```

A "Clear" button is located at the bottom right of the Shell window.

- Basic Python Program Explanation
 - In this example, we start with a comment that explains the purpose of the code. Then, we import the random module to generate random numbers later in the program.
 - Next, we define three variables with different data types: x is an integer, y is a string, and z is a boolean. We then perform an arithmetic operation on x and store the result in the result variable.
 - We use an if-else statement to check the value of z. If z is True, we print the string stored in y, otherwise, we print the value of result.
 - Finally, we define a function called greet() that takes a parameter called name and prints a greeting message. We call the function with the argument "Alyanna" to print "Hello, Alyanna!" to the console.