COURSE CODE: PRAC3301 Section: BSIT

COURSE TITLE: OJT/PRACTICUM 1

TERM: 1st Semester, SY 2024-2025

PROFESSOR: MR. VINCENT CARLO T. GARADOS

DATE: Nov, 2, 2024

NAME OF STUDENT: Jaybie E. Sosmeña

SPEAKERS NAME: Sir. Marc Anthony Tonido

TITLE: Fundamentals of Cybersecurity

**OJT NARRATIVE REPORT**

1. SCREENSHOT OF DUTY
2. NARRATIVE REPORT

The webinar began with an **introduction to Python**, explaining why it has become one of the most popular languages for AI. Python’s simplicity and versatility were highlighted as key reasons for its widespread adoption, along with its extensive ecosystem of libraries like TensorFlow, PyTorch, and scikit-learn. The session emphasized how Python's ease of learning makes it an ideal starting point for beginners while offering enough power and flexibility for advanced AI applications. I also learned about Python’s ability to integrate with other languages and its strong community support, which ensures ample resources and documentation for learners.The next part of the session covered **basic Python syntax**, which laid the foundation for writing clean and effective code. I found the explanation of Python’s indentation rules particularly interesting, as it uses spaces or tabs to define blocks of code rather than braces {} like other languages. This design choice enhances readability and encourages good coding practices.The use of **comments** in Python was also discussed, with a focus on how they begin with the # symbol and are used to document code for better understanding. Additionally, the webinar explained Python’s operators, including arithmetic operators like +, -, and \*, comparison operators such as == and !=, and logical operators like and, or, and not. These fundamentals are essential for performing calculations, comparisons, and logical operations in any program.An interesting segment of the session introduced **modules and packages**, which allow Python code to be organized into manageable pieces. Modules are individual Python files, while packages are collections of modules. This organizational feature simplifies complex projects and enables developers to reuse code effectively.The webinar moved on to explore **data types and variables**, showcasing Python’s dynamic typing. I learned that in Python, there is no need to explicitly declare variable types; the language automatically assigns types based on the values provided. For instance, integers are used for whole numbers, floats for decimals, strings for text, and booleans for true/false values.The discussion also delved into Python’s **data structures**, which are powerful tools for managing collections of data. I was particularly intrigued by **Lists**, which are ordered and mutable collections.**Tuples**, which are similar to lists but immutable.**Dictionaries**, which store key-value pairs where keys are unique. **Sets**, which are unordered collections of unique elements.These data structures are versatile and can be used to solve a wide range of problems efficiently. Additionally, the session touched on Python’s support for **object-oriented programming (OOP)**, enabling developers to create and manage classes and objects for more structured and modular code. The final part of the webinar focused on **control flow**, which allows developers to write conditional statements and loops to guide program execution. The use of if, elif, and else statements was thoroughly explained, showing how Python can evaluate conditions and execute corresponding code. This segment emphasized how control flow is essential for building intelligent and responsive programs.