

LABORATORY MANUAL

Object-Oriented Programming (CPE 103)

|  |  |
| --- | --- |
| **Laboratory Activity No. 2.2** | |
| **Literals, Operators, and Variables** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** 01/25/2025 |
| **Section:** 1 - A | **Date Submitted:** 01/25/2025 |
| **Name:** Eulin, Ryan Bertrand B. | **Instructor:** Maria Rizette M. Sayo |
| **1. Objective(s):** | |
| 1. Implement literals and variables in a python program. | |
| **2. Intended Learning Outcomes (ILOs):** | |
| The students should be able to:   * 1. Write a simple program implementing literals and variables   2. Use comments and identify keywords from identifiers created by users. | |
| **3. Discussion:** | |
| The role of variables, constants, and literals in a Python program is to be able to store information in the program which allows the programmer to manipulate and manage data in order to perform certain tasks. | |
| **4. Materials and Equipment:** | |
| Desktop Computer with Anaconda Python /Python Colab Windows Operating System | |
| **5. Procedure:** | |
| **Perform the activity using the Jupyter Notebook**  This activity can be done either locally on Anaconda’s Jupyter Notebook or online through Google Collaboratory which offers a free Jupyter Notebook environment for Google Users. IPython Notebook files (.ipynb) that are saved in the Google Drive can be opened on Google Collaboratory. Additional guides are available on the IPython Notebook template file that is provided wit h  this activity. If the template is not present, these are the valuable links for reference: | |

|  |
| --- |
| <https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html> <https://colab.research.google.com/notebooks/welcome.ipynb> <https://colab.research.google.com/notebooks/markdown_guide.ipynb>  **Using Variables to Assign Data Types in Python**   1. Input the name of the student.      1. Input the prelim grade and calculate the prelim class standing and total prelim grade.      1. Input the midterm grade and calculate the midterm class standing and total midterm grade.      1. Input the final grade and calculate the final class standing and total final grade.      1. Convert the final grade to the UCC's numerical grade system. |
| **6. Supplementary Activity:** |
| **Tasks**   1. **Test 3 students from the program you created.** 2. **The program should show the name of the student, the PRELIM, MIDTERM, and FINAL grades.** 3. **Convert the final grade into the UCCs numerical grade. Please refer to the grading system.**   For 1, 2 and 3. Please refer to this link: <https://colab.research.google.com/drive/1f_8i2Yb3RjyPqg7KtvyZOun248OsAqxN>  **Questions:**   * 1. **What is the purpose of this Python program?**   The purpose of the this python program is to calculate the grades of 3 students in prelim,  Midterm Final and also to, convert the final grade to the UCC Conversion Grade.   * 1. **How were variables and literals used in this program?**   The program uses variable to store user inputs, calculates grades, and results, while literals define fixed values like grading weight and menu options. |

|  |
| --- |
| **7. Conclusion:** |
| Overall, this python program helps to calculate the grade of the 3 students for their Prelim, Midterm and Final Grade as well as converting the Final Grade to UCC Grading System. It uses variables to store data and literals for fixed values, making the calculations accurate and easy to follow. |
| **8. Assessment Rubric:** |