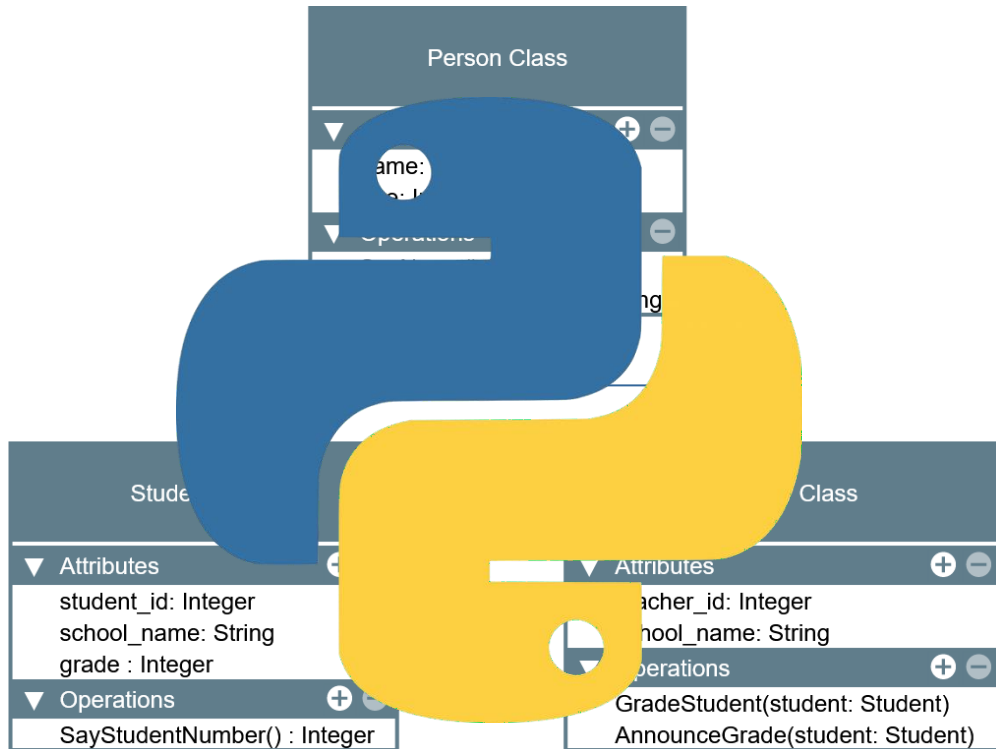




**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*



# LABORATORY MANUAL

## Object-Oriented Programming (CPE 103)



**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*

### Laboratory Activity No. 2.2

#### Literals, Operators, and Variables

**Course Code:** CPE103

**Program:** BSCPE

**Course Title:** Object-Oriented Programming

**Date Performed:** 01/02/2025

**Section:** 1A

**Date Submitted:** 08/02/2025

**Name:** Disomnong, Jalilah M.

**Instructor:** Engr. Maria Rizette H. Sayo

#### 1. Objective(s):

This activity aims to be familiarize in the various data types of Python, assign values to variables, and perform operations in a Python program.

#### 2. Intended Learning Outcomes (ILOs):

The students should be able to:

- Perform different operations available with variables in Python

#### 3. Discussion:

The basis of a programming language contains data elements and the block in which they are being stored. Specific names are given for each of these, and special functionalities can be performed on them. In a programming language, they are called as Variables, Constants, and Literals. In this article, we will look at Python Constants, Variables, and Literals along with their types and examples. A Variable is a location that is named in order to store data while the program is being run. In a programming language, Variables are words that are used to store values of any data type. A Python Constant is a variable whose value cannot be changed throughout the program. Certain values are fixed and are universally proven to be true. These values cannot be changed over time. Such types of values are called as Constants. We can think of Python Constants as a bag full of fruits, but these fruits cannot be removed or changed with other fruits.

Note – Unlike other programming languages, Python does not contain any constants. Instead, Python provides us a Capitalized naming convention method. Any variable written in the Upper case is considered as a Constant in Python.

In Python, there are 4 different types of Literal Collections. They represent more complicated and complex data and assist Python scripts to be more extensible.

1. List Literals The elements in a list are of many data types.
2. Tuple Literals Just like a List, a tuple is also a collection of various data types. It is surrounded by parentheses '()', and each element is separated by a comma (,). It is unchangeable (immutable).
3. Dict Literals The data is stored in the dictionary as a key-value pair. It is surrounded by curly braces '{}', and each pair is separated by commas (,). A dictionary can hold various types of data. Dictionaries are subject to change.
4. Set Literals Set is an unordered data set collection. It is surrounded by and each element is separated by a comma (,).

#### 4. Materials and Equipment:

Desktop Computer with Anaconda Python /Python Colab  
Windows Operating System



**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*

## **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 with this activity. If the template is not present, these are the valuable links for reference:

1. A teacher wants to calculate the final grade in a CpE course and want to write it in a python program. The following are the requirements:
  1. PRELIM GRADE = 50% Prelim Exam + 50% Prelim Class Standing (CS)
  2. PRELIM CS = 50% Hands-on activity + 30% Quiz + 20% Assignment
  3. MIDTERM GRADE = 1/3 of PRELIM GRADE + 2/3 of (50% Midterm Exam + 50% Midterm Class Standing (CS))
  4. MIDTERM CS = 50% Hands-on activity + 30% Quiz + 20% Assignment
  5. FINAL GRADE = 1/3 of MIDTERM GRADE + 2/3 of (50% Final Exam + 50% Final Class Standing (CS))
  6. FINAL CS = 50% Hands-on activity + 30% Quiz + 20% Assignment
  7. HOAs, Quizzes and Assignments are inputted as average of all submissions and are out of 100%.
  8. Major exams are inputted out of 100%.
  9. Show the codes that successfully run the program.
  10. Provide comments or documentation strings for your program.



**UNIVERSITY OF CALOOCAN CITY**  
Caloocan, 1400 Metro Manila, Philippines

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
2<sup>nd</sup> Semester, School Year 2024-2025

```
1 class StudentGrade:
2     def __init__(self, name):
3         self.name = name
4         self.prelim_exam_score = 0
5         self.midterm_exam_score = 0
6         self.final_exam_score = 0
7         self.activity_scores = []
8         self.quiz_scores = []
9         self.assignment_scores = []
10
```

```
11 def input_scores(self):
12     print("\nPrelim Grade")
13     # Prelim Exam Score
14     self.prelim_exam_score = float(input("Enter prelim score exam: "))
15
16     # Activities, Quizzes, and Assignments for Prelim
17     self.activity_scores = [
18         float(input("Activity No.1 Score: ")),
19         float(input("Activity No.2 Score: ")),
20         float(input("Activity No.3 Score: "))
21     ]
22     self.quiz_scores = [
23         float(input("Quiz No.1 Score: ")),
24         float(input("Quiz No.2 Score: "))
25     ]
26     self.assignment_scores = [
27         float(input("Assignment No. 1 Score: ")),
28         float(input("Assignment No. 2 Score: "))
29     ]
30
31     print("\nMidterm Grade")
32     # Midterm Exam Score
33     self.midterm_exam_score = float(input("Enter midterm score exam: "))
34
35     # Activities, Quizzes, and Assignments for Midterm
36     self.activity_scores += [
37         float(input("Activity No.1 Score: ")),
38         float(input("Activity No.2 Score: ")),
39         float(input("Activity No.3 Score: "))
40     ]
41     self.quiz_scores += [
42         float(input("Quiz No.1 Score: ")),
43         float(input("Quiz No.2 Score: "))
44     ]
```



**UNIVERSITY OF CALOOCAN CITY**  
Caloocan, 1400 Metro Manila, Philippines

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
2<sup>nd</sup> Semester, School Year 2024-2025

```
44         ]
45     self.assignment_scores += [
46         float(input("Assignment No. 1 Score: ")),
47         float(input("Assignment No. 2 Score: "))
48     ]
49
50     print("\nFinal Grade")
51     # Final Exam Score
52     self.final_exam_score = float(input("Enter final score exam: "))
53
54     # Activities, Quizzes, and Assignments for Final
55     self.activity_scores += [
56         float(input("Activity No.1 Score: ")),
57         float(input("Activity No.2 Score: ")),
58         float(input("Activity No.3 Score: "))
59     ]
60     self.quiz_scores += [
61         float(input("Quiz No.1 Score: ")),
62         float(input("Quiz No.2 Score: "))
63     ]
64     self.assignment_scores += [
65         float(input("Assignment No. 1 Score: ")),
66         float(input("Assignment No. 2 Score: "))
67     ]
```

```
69     def calculate_grades(self):
70         # Prelim Grade Calculation
71         prelim_activity_avg = sum(self.activity_scores[:3]) / 3
72         prelim_quiz_avg = sum(self.quiz_scores[:2]) / 2
73         prelim_assignment_avg = sum(self.assignment_scores[:2]) / 2
74
75         prelim_cs = (prelim_activity_avg * 0.5) + (prelim_quiz_avg * 0.3) +
76                     (prelim_assignment_avg * 0.2)
77         prelim = self.prelim_exam_score * 0.5
78         self.prelim_grade = prelim + (prelim_cs * 0.5)
79
80         # Midterm Grade Calculation
81         midterm_activity_avg = sum(self.activity_scores[3:6]) / 3
82         midterm_quiz_avg = sum(self.quiz_scores[2:4]) / 2
83         midterm_assignment_avg = sum(self.assignment_scores[2:4]) / 2
84
85         midterm_cs = (midterm_activity_avg * 0.5) + (midterm_quiz_avg * 0.3) +
86                     (midterm_assignment_avg * 0.2)
87         midterm = self.midterm_exam_score
88         self.midterm_grade = (midterm_cs * 0.5) + (midterm * 0.5)
89
90         # Final Grade Calculation
91         final_activity_avg = sum(self.activity_scores[6:9]) / 3
92         final_quiz_avg = sum(self.quiz_scores[4:6]) / 2
93         final_assignment_avg = sum(self.assignment_scores[4:6]) / 2
94
95         final_cs = (final_activity_avg * 0.5) + (final_quiz_avg * 0.3) +
96                     (final_assignment_avg * 0.2)
97         final = self.final_exam_score
98         self.final_grade = (final_cs * 0.5) + (final * 0.5)
99
100        # Midterm and Final Grades Calculation
101        midterm121 = (self.prelim_grade * 0.3) + (self.midterm_grade * 0.7)
102        self.final_grade_121 = (midterm121 * 0.3) + (self.final_grade * 0.7)
```



**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*

```
101 ▾ def assign_letter_grade(self):
102     # Letter Grade Assignment based on final grade
103 ▾     if self.final_grade_121 >= 98:
104         return "Grade: 1.0"
105 ▾     elif self.final_grade_121 >= 95:
106         return "Grade: 1.25"
107 ▾     elif self.final_grade_121 >= 92:
108         return "Grade: 1.5"
109 ▾     elif self.final_grade_121 >= 90:
110         return "Grade: 1.75"
111 ▾     elif self.final_grade_121 >= 86:
112         return "Grade: 2.0"
113 ▾     elif self.final_grade_121 >= 85:
114         return "Grade: 2.25"
115 ▾     elif self.final_grade_121 >= 80:
116         return "Grade: 2.50"
117 ▾     elif self.final_grade_121 <= 60:
118         return "Grade: 3.0"
119 ▾     else:
120         return "Grade: 5.0"
121
122 ▾ def display_results(self):
123     print(f"\nPrelim Grade: {self.prelim_grade}")
124     print(f"Midterm Grade: {self.midterm_grade}")
125     print(f"Final Grade: {self.final_grade_121}")
126     print(self.assign_letter_grade())
127
```

refer to this link :

[https://colab.research.google.com/drive/12VO2A2wVtYaqAWnm-tOwfUa\\_2AvWPmuk](https://colab.research.google.com/drive/12VO2A2wVtYaqAWnm-tOwfUa_2AvWPmuk)



**UNIVERSITY OF CALOOCAN CITY**  
Caloocan, 1400 Metro Manila, Philippines

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
2<sup>nd</sup> Semester, School Year 2024-2025

## 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.

Enter your name: Jalilah

Prelim Grade

Enter prelim score exam: 98

Activity No.1 Score: 78

Activity No.2 Score: 89

Activity No.3 Score: 70

Quiz No.1 Score: 80

Quiz No.2 Score: 80

Assignment No. 1 Score: 90

Assignment No. 2 Score: 70

Midterm Grade

Enter midterm score exam: 80

Activity No.1 Score: 89

Activity No.2 Score: 90

Activity No.3 Score: 87

Quiz No.1 Score: 88

Quiz No.2 Score: 88

Assignment No. 1 Score: 100

Assignment No. 2 Score: 100

Final Grade

Enter final score exam: 87

Activity No.1 Score: 90

Activity No.2 Score: 90

Activity No.3 Score: 90

Quiz No.1 Score: 87

Quiz No.2 Score: 90

Assignment No. 1 Score: 95

Assignment No. 2 Score: 95

Prelim Grade: 88.75

Midterm Grade: 85.36666666666667

Final Grade: 88.05699999999999

Grade: 2.0

Enter your name: Jenny

Prelim Grade

Enter prelim score exam: 100

Activity No.1 Score: 90

Activity No.2 Score: 90

Activity No.3 Score: 90

Quiz No.1 Score: 90

Quiz No.2 Score: 90

Assignment No. 1 Score: 90

Assignment No. 2 Score: 90

Midterm Grade

Enter midterm score exam: 95

Activity No.1 Score: 95

Activity No.2 Score: 95

Activity No.3 Score: 95

Quiz No.1 Score: 95

Quiz No.2 Score: 95

Assignment No. 1 Score: 100

Assignment No. 2 Score: 100

Final Grade

Enter final score exam: 100

Activity No.1 Score: 80

Activity No.2 Score: 100

Activity No.3 Score: 100

Quiz No.1 Score: 100

Quiz No.2 Score: 100

Assignment No. 1 Score: 90

Assignment No. 2 Score: 90

Prelim Grade: 95.0

Midterm Grade: 95.5

Final Grade: 96.73833333333332

Grade: 1.25

Enter your name: Jolly

Prelim Grade

Enter prelim score exam: 80

Activity No.1 Score: 80

Activity No.2 Score: 80

Activity No.3 Score: 87

Quiz No.1 Score: 75

Quiz No.2 Score: 88

Assignment No. 1 Score: 87

Assignment No. 2 Score: 89

Midterm Grade

Enter midterm score exam: 95

Activity No.1 Score: 60

Activity No.2 Score: 60

Activity No.3 Score: 80

Quiz No.1 Score: 87

Quiz No.2 Score: 89

Assignment No. 1 Score: 78

Assignment No. 2 Score: 67

Final Grade

Enter final score exam: 90

Activity No.1 Score: 90

Activity No.2 Score: 89

Activity No.3 Score: 87

Quiz No.1 Score: 89

Quiz No.2 Score: 90

Assignment No. 1 Score: 90

Assignment No. 2 Score: 90

Prelim Grade: 81.60833333333333

Midterm Grade: 84.61666666666667

Final Grade: 87.82841666666667

Grade: 2.0





**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*

### Questions:

1. What are the key components that contribute to the final grade in the CpE course, and how is each component weighted in the overall calculation?

The final grade for a CpE course is typically determined by a combination of different components, such as the prelim exam, the midterm exam, assignments, quizzes, and participation, which make up the remaining percentage.

2. Explain the purpose of using functions in the grade calculation program. How do they improve the structure and readability of the code?

Functions play a vital role in structuring the grade calculation program. They enhance the readability, organization, and maintainability of the code by breaking it down into manageable, self-contained sections. Each function performs a specific task, such as calculating grades, converting numeric scores into letter grades, or handling user input.

3. Describe the process of calculating the prelim grade. What specific scores are considered, and how are they combined to arrive at the final prelim grade?

The process of calculating the prelim grade typically involves gathering various scores from the student, such as quiz scores, midterm exam results, and assignment scores. These scores are then assigned specific weights based on their importance in the course.

4. What potential errors could arise from user input in this program, and how would you implement error handling to ensure that the scores entered are valid (e.g., within a specific range)?

When it comes to user input, several potential errors could arise, such as entering non-numeric values (e.g., letters instead of numbers), inputting scores outside the valid range (such as greater than 100 or less than 0), or leaving the input blank. To a

5. How does the program determine the letter grade based on the final numeric score? What are the specific ranges for each letter grade, and how are they implemented in the code?

Once the final grade is calculated, the program will convert this grade into a numeric based on predefined ranges. The program typically uses conditional statements (if-elif-else) to check the final numeric score and assign the appropriate letter grade.





**UNIVERSITY OF CALOOCAN CITY**  
*Caloocan, 1400 Metro Manila, Philippines*

**COLLEGE OF ENGINEERING**  
**Computer Engineering**  
*2<sup>nd</sup> Semester, School Year 2024-2025*

## **7. Conclusion:**

This laboratory exercise provided valuable practical experience with fundamental Python programming concepts, including literals, variables, and operators. By developing a program for grade calculation, we enhanced our understanding of the necessity for structure and modularity in coding through the use of functions. Functions facilitate the decomposition of intricate tasks, thereby improving code readability and maintainability. Furthermore, the activity allowed us to engage in input validation and error handling, which ensures that users can submit valid data for grade computations. We also examined the process of calculating grades, considering various elements such as exams, assignments, and quizzes. This exercise highlighted the importance of logical reasoning in programming and the application of conditional statements for grade conversion. In summary, this activity not only deepened our grasp of essential programming principles but also illustrated how these principles can be utilized in practical contexts, such as academic grading systems.

## **8. Assessment Rubric:**