

**SAINT VINCENT COLLEGE OF CABUYAO**

***Bachelor of Science in Information Technology***

**LABORATORY MANUAL**

**PF101 – Object-Oriented Programming**

**Laboratory Exercise No. 3**

***Class Fields and Methods***

Submitted by:

|  |
| --- |
| **Magno, Lancelei C.** |

[Name]

|  |
| --- |
| **BSIT-2A2** |

[Section]

Submitted to:

|  |
| --- |
| **Prof. Apollo Neil R. Duran** |

[Name of Instructor/Professor]

|  |
| --- |
|  |

**GRADE**

|  |
| --- |
| **AUG 2024** |

[Date]



***Laboratory Exercise No. 3***

**Class Fields and Methods**

1. OBJECTIVES

|  |
| --- |
| **At the end of the exercise, the students are expected to:**   * **Declare instance variables in a program** * **Use class fields in a program** * **Create a method in a program** |

1. EQUIPMENT/ MATERIALS

|  |
| --- |
| **The following equipment or materials will be needed to perform the laboratory exercise:**   * **PC with Java Compiler and IDE (Eclipse, NetBeans, jGrasp, etc.)** * **Internet Connection for Online Java Compiler/Editor and Submission** * **USB for backup and file storage** |

1. PROCEDURE/ DISCUSSION

|  |
| --- |
| **Laboratory Work No. 1**  From the given image below, create your own Java program to illustrate a class named Dog.java that will contain the given set of class fields. A class method of the Dog class named *“ShowDetails()”* will be called inside the main method through an object of the Dog class named *“Rayne”* in order to print the all the contents of the class fields of the object in the console window. A separate class named *“Main.java”* must contain the main() method that will execute the code. |

|  |
| --- |
| **Laboratory Work No. 2**  Construct a Java program named PF101LabExer3-2.java whose main() methods must hold two integer variables. Then, assign a value to the variables. Pass both variables to methods named sum() and difference(); these compute the sum and difference between the values of two arguments, respectively. Each method should perform the appropriate computation and display the results. |

1. **DATA REPRESENTATION/ OUTPUT PICTURES**

|  |
| --- |
| **~LABWORK 1~**      **~LABWORK 2~** |

1. RESULTS INTERPRETATION/ OBSERVATION

|  |
| --- |
| You can create a class in lab work number 1 with base properties that you can later modify to its predefined object values. Since the height and length attributes are not allowed to exceed a value of 127, they were set to byte data type to minimize the amount of memory consumed. Additionally, as the weight cannot exceed 32,000, it is assigned to short data type. Since they are a collection of characters, the remaining ones were set to String.  Lab work 3 utilizes methods to make a predefined task that you can later use. I have noticed throughout my programming experience, some methods from other libraries that I use have the same method name but takes different data types so I tried it on mine. I have come to realize that sum and difference doesn't only occur in int data type but also on other so I created methods that could take those. |

1. CONCLUSIONS

|  |
| --- |
| OOP makes the code much easier to read and understand by dividing it into separate sections. If a class was giving you trouble, all you would have to do would be to go through the file, look it up, and troubleshoot. Furthermore, managing and altering objects is easier.  Multiple methods with the same job but various data types of parameters were made easy to have by overloading. This kind of prebuilt code might greatly improve your programming experience. |

1. STUDENT OUTCOMES ADDRESSED

|  |
| --- |
| ***(… to fill out by your instructor)*** |

1. APPENDICES
   1. RUBRICS AND SCORING

|  |
| --- |
| ***(… kindly refer to rubrics and scoring provided)*** |