

**SAINT VINCENT COLLEGE OF CABUYAO**

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

**LABORATORY MANUAL**

**PF101 – Object-Oriented Programming**

**Laboratory Exercise No. 2**

***Classes and Objects***

Submitted by:

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

[Name]

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

[Section]

Submitted to:

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

[Name of Instructor/Professor]

|  |
| --- |
|  |

**GRADE**

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

[Date]



***Laboratory Exercise No. 2***

**Classes and Objects**

**I. OBJECTIVES**

|  |
| --- |
| **At the end of the exercise, the students are expected to:**   * **Declare class and variables in a program** |

**II. 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** |

**III. PROCEDURE/ DISCUSSION**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Laboratory Work No. 1 Create a class named “Lot” having class fields such as the address, ownerFirstName, ownerLastName, length and width. Determine the most appropriate data type for each field through the given set of default class field values:   |  |  | | --- | --- | | **Class Field** | **Default Value** | | address | Cabuyao | | ownerFirstName | Juan | | ownerLastName | Dela Cruz | | lotLength | 5.60 | | lotWidth | 7.80 |   A class named “MainProgram” will be defined in order to execute the program by including the given code from the given items as the body of the main method of the class “MainProgram” one after another. By doing so, you can answer the following items by writing your answers on the space provided. If the statements will result to a compile time or runtime error, do not write anything except ERROR as your answer. Laboratory Work No. 2Treat each number as part of new code execution, thus will not affect the changes in their respective values. |

|  |
| --- |
| 1. Lot objLot = new Lot();   System.out.println(“Lot’s address : “ + objLot.address);   1. Lot objLot = new Lot();   System.out.println(“Owner’s Full name : “ + objLot.ownerFirstName + “ “ + objLot.ownerLastName);   1. Lot objLot = new Lot();   System.out.println(“Lot Area : “ + (objLot.length \* objLot.width));   1. Lot objLot = new Lot();   System.out.println(“Old Lot Area : “ + (objLot.length \* objLot.width));  objLot.length = 10;  System.out.println(“New Lot Area : “ + (objLot.length \* objLot.width));   1. Lot objLot = new Lot();   objLot.address = Cabuyao, Laguna;  System.out.println(“Lot’s address : “ + objLot.address)   1. Lot objLot = new Lot();   objLot.ownerFirstName = “John”;  objLot.ownerLastName = “Jones”;  System.out.println(“Hello ” + objLot.ownerFirstName + “!”)   1. Lot objLot = new Lot();   objLot.width = “8”;  System.out.println(“Lot Area : “ + (objLot.length \* objLot.width));   1. Lot objLot1 = new Lot();   Lot objLot2 = new Lot();  objLot2.length = objLot1.length \* 3;  objLot2.width = objLot1.width + objLot1.length;  System.out.println(“Lot Area of Lot 2: “ + (objLot2.length \* objLot2.width));   1. Lot objLot1 = new Lot();   Lot objLot2 = new Lot();  System.out.println(“Owner’s Full name of Lot 1: “ + objLot1.ownerFirstName + “ “ + objLot1.ownerLastName);  System.out.println(“Owner’s Full name of Lot 2: “ + objLot2.ownerFirstName + “ “ + objLot2.ownerLastName);   1. Lot objLot1 = new Lot();   Lot objLot2 = new Lot();  objLot.address = “Calamba”;  System.out.println(“Lot 1 address : “ + objLot1.address);  System.out.println(“Lot 2 address : “ + objLot2.address); Laboratory Work No. 3Construct a Java code to employ a class named Students, then initialize attributes like *name*, *address*, *birthday*, *age*, *email*, and *grade* while creating an object named BSIT. The given attributes must be inputted by user with the help of the *BufferedReader*. Save your program as PF101LabExer2-3.java. |

**IV. DATA REPRESENTATION/ OUTPUT PICTURES**

|  |
| --- |
| **~LABWORK WORK 1~**      **~LABWORK WORK 3~** |

**V. RESULTS INTERPRETATION/ OBSERVATION**

|  |
| --- |
| The first lab work gave you access to a class with default values that you may change at a later time. It will always return the class's default values if you call it without modifying the value.  Instead of using a Scanner, the third lab work employed a buffered reader to obtain input. Given that BufferedReader reads buffered characters in your data, it is technically considerably quicker than Scanner. I utilized a modified version of BufferedReader in my work, which uses the constructor InputStreamReader as the source. It is an improved version of the Scanner(System.in) that we often utilize. |
| ***Results of Laboratory Work No. 2 here…*** |
| **1.** |
| **2.** |
| **3.** |
| **4.** |

|  |
| --- |
| **5.** |
| **6.** |
| **7.** |
| **8.** |
| **9.** |
| **10.** |

**VI. CONCLUSIONS**

|  |
| --- |
| OOP greatly simplifies data manipulation, particularly when dealing with data that has a base value such as pi that may be modified at any moment.  With the introduction of BufferedReader, we learned another tool that can make our programming experiences more efficient. |

**VII. STUDENT OUTCOMES ADDRESSED**

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

**VIII. APPENDICES**

1. **RUBRICS AND SCORING**

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