WEEK 1 Introduction to Object-Oriented Programming
Hands-on Coding Exercise

Group Members (LN, FN MI.): In alphabetical Order	1			**Score:**	
	2			**Date:**	
Section:		**Instructor:**	Prof. Janus Raymond C. Tan	**Sem./ A.Y.**	1st/ 23-24
Project Link:	*Paste here the link of your project uploaded in your account from* ***https://codiva.io/****. Double check if the link is accessible. Do not share your project link to others to avoid copying.* ***Make your own secured and extremely random Project Name so that it will not be easy for other groups to search your project once uploaded in https://codiva.io/.***				

CRITERIA	GRADING SCALE				WEIGHT	SCORE
	NEEDS IMPROVEMENT 1	FAIR 2	GOOD 3	EXCELLENT 4		
CODE PROFICIENCY	Code shows minimal understanding of programming concepts, with solutions that are ineffective or incorrect.	Code demonstrates a basic understanding of concepts, but solutions may lack efficiency or elegance.	Code shows a solid grasp of programming concepts, and solutions are effective.	Code demonstrates exceptional understanding and mastery of programming concepts. Efficient and elegant solutions are consistently employed.	30%	
FUNCTIONALITY	The program does not meet essential requirements and has significant functionality problems or errors.	The program partially meets requirements, but functionality may be limited or inconsistent. Several bugs or issues affect program behavior.	The program meets most specified requirements and functions correctly in typical scenarios. Minor bugs or issues may be present.	The program fully meets all specified requirements and functions flawlessly under various scenarios.	30%	
DOCUMENTATION	Documentation is absent or provides little to no insight into code logic or usage.	Documentation is minimal and lacks clarity in explaining code logic or usage. Important details are missing.	Documentation is present and adequately explains code logic and usage. Some details may be lacking.	Comprehensive and well-organized documentation is provided, including clear explanations of code logic, usage, and any assumptions made.	20%	
CODE READABILITY	Code lacks organization and readability due to unclear variable names, inadequate comments, or messy format	Code is somewhat organized, but variable names, comments, or formatting may be inconsistent or unclear.	Code is well-organized, with clear variable names and comments. Formatting is consistent, aiding in understanding.	Code is exceptionally well-organized, with meaningful variable names, consistent formatting, and clear comments. Easy to understand and maintain.	20%	
				TOTAL	100%	

INSTRUCTION:

In this exercise, you will create a basic Java class that represents an object from your daily life. This is an opportunity to practice creating classes, defining attributes, and writing methods in Java.

1. Choose an everyday object around you. Think about the attributes (class fields) and behaviors (methods) this object might have.

2. Create a Java class with an appropriate name for the chosen object. The class should have the following components:

a. Class name: Choose a suitable name for your class, following Java naming conventions (CamelCase).

b. Attributes: Identify at least three attributes (class fields) that describe the object. These could be characteristics or properties of the object. Identify also the most appropriate data type for each attribute.

c. Methods: Define at least two methods that represent actions or behaviors associated with the object.

3. Define setter and getter methods for each identified attribute (class fields).

4. Write the method body of the methods you defined. These methods should perform actions related to the object's behaviors.

5. Create an instance of the class (object): In the main method, create an instance of your class and use the setter methods to set its attribute values.

6. Call the getter methods to display the instance attribute values on user's screen including the methods defined to demonstrate the actions associated with your object.

PROGRAM CODE

SAMPLE OUTPUT

WHAT HAVE YOU LEARNED?

Attach your own PDF file **(every group member should turn in his/ her own copy)** as your submission in the assigned task upon turning in using your own account in our LMS following the given template. **(your PDF file should begin with the first page of the given template)**