



PROJECT: Version 01

Course Identification

Name of programs – Codes:	COMPUTER SCIENCE TECHNOLOGYPROGRAMMING – 420.BP INFORMATION TECHNOLOGY PROGRAMMERANALYST - LEA.3Q
Course title:	OBJECT- ORIENTED PROGRAMMING
Course number:	420-CT2-AS
Group:	07030
Teacher's name:	Houria Houmel
Duration:	Project (Version 01)
Semester:	Summer 2022

Student Identification

Name: _____

Student number: _____

Due Date: **July 18, 2022,**

Result: _____

☐ I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work.

Standard of the Evaluated Competency

Statement of the evaluated competency – Code

Use an object-oriented development approach-00Q6

Evaluated elements of the competency

1. Analyze the problem
2. Model the classes
3. Produce the algorithms for methods
4. Create the graphical interface
5. Program the classes

Instructions

- **Due Date: July 18, 2022**
- **Teamwork**
- **S**tudents must present the project and the presentation date will be scheduled for each team.
- Plagiarism: attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. (IPEL – Article 5.16).
- Please submit the project named ***Bike Factory*** via LEA in the Assignment folder named Project.

❖ **Part 1:** The Business Layer

Implementing the business classes

❖ **Part 2:** The Data Layer

Implementing the data file class

❖ **Part 3:** The Presentation Layer

Implementing the Windows Forms Application

Bike Factory

In this evaluation, you will create an application to represent a factory of bikes that manufactures different types of bikes.

Bike Factory manufactures different types of bikes.

A **bike** is identified by a ***serial number***, ***make***, ***model***, ***color***, ***speed***, and ***made date***.

The factory manufactures **mountain bikes** and **road bikes**.

A **mountain bike** has its specific features, such as the ***suspension***; it has three types of suspension (**front**, **rear**, and **front and rear**). You need also to know what the measurement of the ***height from ground*** is.

When you check for a **road bike**, you need to know what the ***seat height*** is.

The list of features is not exhaustive; you need to add one feature to each class.

You have been assigned the job to ***design*** and ***implement*** an application named ***Bike Factory*** with the three layers: business layer (**bus**), data layer(**data**) and presentation layer (**client**).

Part 1. Business Layer:

Question 1

Create an **enumeration** named ***EnumColor***, having the three constants **Red**, **Blue**, and **Dark**.

Question 2

Create an **abstract class** **Bike** with the following attributes: **serial number** of the **long** type, **make** of the **string** type, **model** of the **string** type, **color** of the **EnumColor** type, **speed** of the **double** type, and **date** of the **Date** type (or **DateTime** from the .Net Framework).

The **Bike** class has the following behaviors (methods):

- The method **double GetMaxSpeed()** initializes the speed of **Bike** by 20. Make this method virtual to allow it to be overridden by the *subclasses* of the **Bike** class.
- The method **void SpeedUp (double newSpeed)** is abstract.

Question 3

All mountain bikes and road bikes share common characteristics and behaviors.

Create two subclasses **MountainBike** and **RoadBike**, which inherit from class **Bike**. The **RoadBike** class has an additional attribute of **double** type called **seatHeight**. The **MountainBike** class has an additional attribute of **EnumSuspension** type called **suspension** and another of **double** type called **heightFromGround**.

- Define the overloaded constructors and the properties in each subclass.
- Override the method **ToString()** that returns a string representation of the object.

Question 4

All **mountain bikes** and **road bikes** share common behaviors (functionalities) that are described in the base class **Bike**.

Each subclass implements the abstract method **SpeedUp()**, as follows:

- In the class **MountainBike** and the class **RoadBike**, the overridden method **void SpeedUp (double newSpeed)** must check, by using the method **double GetMaxSpeed ()**, if the (speed + new speed) is lower than the maximum speed then the speed will be changed as follows: **speed += newSpeed**, otherwise the speed will be not changed

Question 5

The class **RoadBike** overrides the *virtual* method **GetMaxSpeed()** of the base class, such as:

- The method **GetMaxSpeed()** initializes the **speed** by **40**

Question 6

Designing the Class Diagram

Design the class diagram using visual studio 2017

Part 2: Data Layer

Create a class named **Data File** to save all the bikes into a file. This class will be used by the client application to save the bikes into a physical file (from the temporary storage to the permanent storage) and, then to load the bikes from this physical file when necessary.

Part 3: Presentation Layer

Managing the stock of bikes

Windows Forms application

Create a **graphic interface**:

To effectively manage the stock of bikes, you must implement the following functions:

- Q1- Create an instance of List<Bike> called listOfBikes
- Q2- Add new bikes (mountain bikes and road bikes) into the list of bikes
- Q3- Search for a bike into the list of bikes by serial number.
- Q4- Update a bike in the list of bikes.
- Q5- Remove a bike from the list of bikes.
- Q6- Print the list of bikes.
- Q7- Print the list of mountain bikes.
- Q8- Print the list of road bikes.
- Q9- Exit the application.