

FORMATIVE ASSESSMENT 3

Project

Programming with C#

PRG521

2024

Table of Contents

Programming with C#	1
PRG521	1
Project-based Questions (100 Marks)	5
Project Tasks:.....	5
Submission Guidelines:	6
Formative Assessment 3 Rubric.....	7

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I _____ hereby

(FULL NAME)

declare that the contents of this assignment are entirely my own work with the exception of the following elements: (List the elements of work in this project that were not self-generated as well as who the originator of the element is)

Element	Originator

Signature: _____ Date: _____

Software Design & Development

Module: Programming with C#

Module Code: PRG521

NQF: 5

Credits: 20

Assessment type: Formative Assessment

Mark allocation: 100

Hand out date: 26 April 2024

Hand in date: 15 May 2024

Instructions:

- Read each question carefully and consider the mark allocation prior to answering.
- Ensure you answer all questions.
- Using any AI tools to complete this assessment is strictly prohibited.
- For final submission, submit a compressed file that includes the Declaration of Authenticity and source code.

Unit Standards:

US Type	US ID	Description	Level	Credits
Core	115392	Apply principles of creating computer software by developing a complete programme to meet given business specifications	NQF-5	12
Core	115373	Demonstrate an understanding of sort and search techniques used in computer programming	NQF-5	6
Core	115359	Demonstrate an understanding of the handling of error in a computer programming environment	NQF-4	2

Assessment outcomes:

- Using structs and enums
- Implementing abstraction
- Implementing encapsulation
- Error and exception handling

Project-based Questions

(100 Marks)

Instructions: Read the scenario and answer the following questions.

Virtual Zoo Management System

You have been tasked to design and implement a Virtual Zoo Management System in C# that simulates the management of animals in a zoo. The system should allow for the addition, tracking, and interaction with various types of animals, showcasing the principles of Object Oriented Programming. This project is a continuation of the Formative Assessment 2.

New Objectives:

- **Using Structs and Enums:** Integrate structs for simple data structures and enums for categorizing data such as animal types, food types, or habitat conditions.
- **Implementing Abstraction:** Utilize abstract classes or interfaces to abstract away complex functionalities into simpler, more general interfaces.
- **Implementing Encapsulation:** Ensure that data is well protected within classes, with strict control over how data is accessed or modified.
- **Error and Exception Handling:** Implement comprehensive error and exception handling to manage and respond to exceptional conditions in a controlled and predictable manner.

Source: Mashile T., (2024)

Project Tasks:

Task 1: Using Structs and Enums

- **Structs:** Define structs for lightweight data structures: `DietInfo` (dietary requirements and feeding schedules).
- **Enums:** Create `enums` to classify information like `AnimalType` (mammal, bird, reptile, etc.), `FoodType` (meat, vegetation, mixed), and `HabitatType` (desert, forest, aquatic).

Task 2: Implementing Abstraction

- Design abstract classes or enhance existing interfaces to represent general animal behaviors, such as an abstract class `Animal` that includes abstract methods like `Eat`, `Move`, and `Speak`.
- Use interfaces to define cross-category capabilities, such as `IClimbable`, `ISwimmable`, or `IFlyable`, depending on the type of animal.

Task 3: Implementing Encapsulation

- Ensure all fields in classes are private and only accessible through public methods or properties.
- Use properties to control the access to class fields with specific `get` and `set` requirements.

Task 4: Error and Exception Handling

- Implement `try-catch` blocks to handle potential runtime errors, such as feeding the wrong type of food to an animal or handling out-of-bounds errors when accessing zoo map positions.
- Use custom exceptions to signal specific errors that can occur in zoo operations, enhancing the clarity of error management.

Additional Requirements:

- Provide a detailed GUI-based interaction system for zookeepers to manage daily tasks.
- Ensure code follows best practices regarding readability, structure, and maintainability.
- Create both:
 - A **ReadMe** text file containing the following: How to run the application, Expected errors, future implementations i.e., conversion to a desktop app or Git push and finally the credits of the developer (you as the student).
 - A **user manual** containing the following: Screenshots of each GUI display, along with labels of the current display and buttons.

Submission Guidelines:

- Submit all source code files, including any base classes, derived classes, and interfaces.
- Include a brief user manual **PDF** file explaining the design decisions, particularly how inheritance, polymorphism, method usage, and interfaces were implemented and utilized in the project.

End of question.

[Total = 100 Marks]

End of paper.

Formative Assessment 3 Rubric

Penalties: Deduct points for significant issues such as lack of proper encapsulation, poor abstraction, ineffective error handling, or any functionality that does not meet the project objectives.

PRG521 FA3 RUBRIC		
Student Name/Number:		
Criterion	Mark Allocation	
	10 – 6	5 – 0
1. Using Structs and Enums (20 Marks)		
Correct Use of Structs (10 Marks): Evaluate how effectively structs are used for appropriate data structures, checking for proper data handling and usage.		
Effective Use of Enums (10 Marks): Assess the use and implementation of enums to improve code readability, maintainability, and the elimination of magic numbers.		
2. Implementing Abstraction (20 Marks)		
Abstraction in Design (10 Marks): Evaluate the design and implementation of abstract classes or interfaces, focusing on how well they generalize functionalities.		
Practical Use of Abstraction (10 Marks): Assess the practical application of abstract methods and interfaces across different classes and scenarios in the project.		
3. Implementing Encapsulation (20 Marks)		
Data Protection (10 Marks): Assess the strictness of access modifiers and the use of properties to encapsulate data within classes.		
Use of Properties (10 Marks): Evaluate the implementation of properties to control data access effectively, ensuring data integrity and validation.		
4. Error and Exception Handling (20 Marks)		
Handling of Expected Errors (10 Marks): Evaluate the handling of common and expected errors via exception-handling mechanisms.		
Custom Exceptions (10 Marks): Assess the creation and use of custom exceptions to clearly identify and manage specific errors related to zoo operations.		
5. Code Quality and Documentation (20 Marks)		
Code Readability and Structure (10 Marks): The code should be clean, well-organized, and easy to follow, with consistent coding standards and logical structuring.		
Comments (5 Marks): Inline comments Documentation (5 Marks): Detailed ReadMe and User manual included.		
TOTAL	/100	