FORMATIVE ASSESSMENT 2

Project

Programming with C#

PRG521

2024

CTU training solutions | Creative & Technology Universitas

Table of Contents

Please complete the declaration of authenticity below for all assignments:

DECLARATION OF AUTHENTICITY

I ____Henrico Jan Christoffel Swanepoel______________________________ 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:** __24/04/2024____

Software Design & Development

Module: Programming with C#

Module Code: PRG521

NQF: 5

Credits: 20

Assessment type: Formative Assessment

Mark allocation: 100

Hand out date: 5 April 2024

Hand in date: 24 April 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:

- Create database access for a computer application using a structured query language.
- Demonstrate an awareness of ethics and professionalism for the computer industry in South Africa.
- Demonstrate fourth-generation language computer programming skills.
- Present information in a public setting.
- Demonstrate an understanding of the principles of the Internet and the World Wide Web.

Assessment outcomes:

- Implementing Inheritance
- Implementing Polymorphism
- Using Methods and Method Overloading
- Using interfaces

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 inheritance, polymorphism, method overloading, and the use of interfaces. This project requires students to create a class hierarchy for animals, implement behaviours through methods and interfaces, and demonstrate polymorphic behaviour.

Objectives:

- Implementing Inheritance: Create a base class for animals and extend it with specific animal classes, showcasing hierarchical relationships.
- Implementing Polymorphism: Utilize polymorphism to allow for different behaviours among animals, such as the way they speak or move, using overridden methods.
- Using Methods and Method Overloading: Demonstrate the use of methods to simulate animal behaviours and interactions, and implement method overloading to handle different scenarios or inputs.
- Using Interfaces: Define and implement interfaces to standardize certain capabilities or features across different animals, such as IFeedable for animals that can be fed or IMovable for animals that can move.

Source: Mashile T., (2024)

Project Tasks:

Task 1: Class Hierarchy and Inheritance

- Design a base class named `Animal` that includes common properties (`e.g., Name, Age`) and methods (`e.g., Eat, Sleep`).
- Extend the `Animal` class to create specific animal classes (`e.g., Lion, Parrot, Turtle`). Each class should override base class methods to provide specific behaviours.

Task 2: Polymorphism and Method Overriding

- Implement method overriding in the derived classes to exhibit unique behaviours. For example, the `Speak` method could return different sounds for different animals.
- Demonstrate polymorphic behaviour by interacting with collections of the base class type (`Animal`) that contain various derived class instances.

Task 3: Methods and Method Overloading

- Implement various methods within animal classes to simulate behaviours (`e.g., Move, Speak`).
- Overload methods to provide different implementations based on parameters. For example, the `Eat` method could be overloaded to accept different types of food.

Task 4: Implementing Interfaces

- Define interfaces such as `IFeedable` with a method `Feed` and `IMovable` with a method `Move`.
- Implement these interfaces in applicable animal classes, ensuring that the behaviours are consistent with the interface definitions.

Additional Requirements:

- Provide a simple GUI interface for interacting with the zoo system, allowing users to add animals, invoke behaviours, and display information about the animals.
- Ensure the application is well-organized and follows best practices for code readability, structure, and comments.

Submission Guidelines:

- Submit all source code files, including any base classes, derived classes, and interfaces.
- Include a brief ReadMe text 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 2 Rubric

Penalties: Points may be deducted for significant issues such as program crashes, major bugs, failing to compile, or not meeting the project objectives.

PRG521 FA2 RUBRIC		
Student Name/Number:		
Criterion	**Mark Allocation**	
	10 – 6	**5 – 0**
1. Implementing Inheritance (20 Marks)		
Base Class Implementation (5 Marks): Evaluate the design of the Animal base class, including the appropriate choice of properties and methods that logically apply to all animals.		
Derived Class Implementation (10 Marks): Assess the creation of specific animal classes that inherit from the Animal base class, looking for a logical hierarchy and the proper use of inheritance.		
Correct Use of Inheritance (5 Marks): Determine if inheritance is used effectively to share code and behaviours among classes, avoiding unnecessary redundancy.		
2. Implementing Polymorphism (20 Marks)		
Method Overriding (10 Marks): Evaluate the use of method overriding in derived classes to provide specific implementations of behaviours, such as how different animals move or speak.		
Polymorphic Behavior (10 Marks): Assess how polymorphism is demonstrated, particularly through the use of base class references to interact with objects of derived classes in a way that exhibits their specific behaviours.		
3. Using Methods and Method Overloading (20 Marks)		
Method Implementation (10 Marks): Evaluate the implementation of methods within classes to simulate animal behaviours and interactions, looking for clear, logical, and efficient code.		
Method Overloading (10 Marks): Assess the use of method overloading to handle different scenarios or inputs, ensuring that overloaded methods provide tangible benefits in flexibility or functionality.		
4. Using Interfaces (20 Marks)		
Interface Design (10 Marks): Evaluate the design of interfaces, such as IFeedable and IMovable, to ensure they provide a meaningful contract for implementing classes.		
Interface Implementation (10 Marks): Assess the implementation of interfaces in classes, looking for consistency with the interface definitions and effective use of interfaces to promote code reusability and flexibility.		
5. Code Quality and Documentation (20 Marks)		
Code Readability and Structure (5 Marks): The code should be well-organized, with meaningful variable and method names, consistent indentation, and clear separation of concerns.		
Comments and Documentation (5 Marks): Inline comments and documentation should accurately explain the purpose and logic of the code, making it accessible to others.		
Design Principles (5 Marks): Evaluate the application of object-oriented design principles, such as encapsulation, abstraction, and the DRY (Don't Repeat Yourself) principle.		
Innovative Features (5 Marks): Extra Marks can be awarded for creativity and the implementation of additional features that enhance the project, such as advanced animal interactions, environmental effects, or graphical user interfaces (GUIs).		
TOTAL	**/100**	