|  |  |
| --- | --- |
| Name: Jiego Pasamanero  Student #: 438585 | |
| **SODV 1202**  **Introduction to Object-Oriented Programming**  Winter 2025  Assignment-02 | |
| **Instructor:** Mahbub Murshed | **Date:** March 25, 2025 |
| **Instructions:**   * Written questions can be submitted as scanned documents or as a softcopy submitted in the d2l Dropbox. Answers should be written clearly and coherently. Programming questions can be submitted using .cs format or compressed zip or rar format. * Assignments must be submitted by the posted due date, or be subject to the course’s late policy. Work must be completed individually and following the academic integrity policy. * For all written questions, you may assume that any necessary using directives are included. | |
| |  |  | | --- | --- | | **Written Questions** | | | **1** | / 10 | | **2** | / 10 | | **3** | / 10 | | **Programming Questions** | | | **Quality** | / 20 | | **Functionality** | / 50 | | **Total** | | | / 100 | | | |

**Assignment 2: Written Questions**

|  |  |  |
| --- | --- | --- |
| 1. | Give concise answers to the following questions | / 10 |
| a) | What is Encapsulation?  Encapsulation is when fields or methods are made inaccessible to unwanted access and combine the methods and fields into a class. | / 5 |
| b) | What are the differences between a class and an abstract class?  Abstract classes cannot initialize or hold objects and any regular class that inherits it must override the methods  Classes can actually hold code and objects while also overrideing any of the methods from an abstract class | / 5 |
| 2. | Design different classes in an inheritance hierarchy/relationship to represent different shapes (circle, square, triangle, rectangle, rhombus, and pentagon). All the shape classes should inherit from a common base class. The classes should contain functions to calculate the shape’s area. Include any other relevant members you think may be needed. (Note that, in this task, you do not need to implement any method, only a class design diagram (or UML) with member fields, properties, and method signatures can satisfy this question.)  A black background with white text  AI-generated content may be incorrect. | / 10 |

|  |  |  |
| --- | --- | --- |
| 1. What is the exact output of the program below?  |  | | --- | | public class Base  {  public Base()  {  Console.WriteLine("Base Constructor");  }  public virtual void Func()  {  Console.WriteLine("Base Func");  }  }  public class Derived : Base  {  public Derived()  {  Console.WriteLine("Derived Constructor");  }  public override void Func()  {  Console.WriteLine("Derived Func");  }  }  class Program  {  static void Main(string[] args)  {  var b = new Base();  b.Func();  var d = new Derived();  d.Func();  }  } | | / 10 |

Base Constructer

Base Func

Base Constructer

Derived Constructer

Derived Func

**Assignment 2: Payroll System Management**

**Overview**

Write a C# program to control the Payroll System of an organization (Application of Polymorphism).

**Directions**

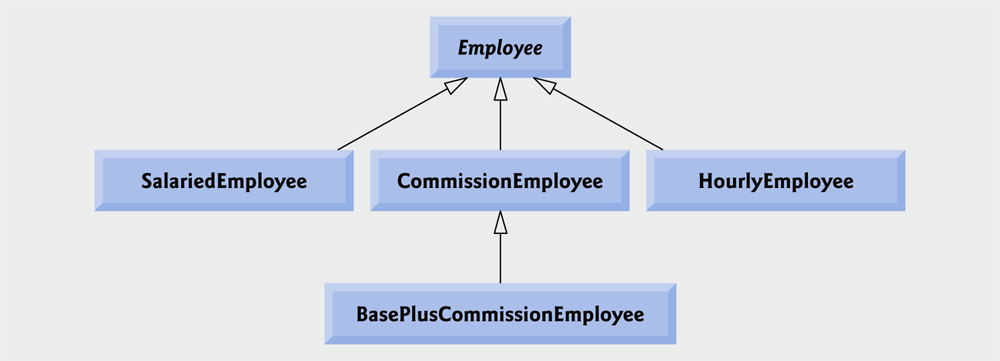
Create an employee class hierarchy (inheritance) in order to solve the following problem: Assume, a company pays its employees every week. There are four different groups of employees.

1. **SalariedEmployee**: Receives fixed weekly payments regardless of the number of hours they work at the workplace.
2. **HourlyEmployee**: Employees are paid by the number of hours they work. They also receive overtime pay for working over 40 hours. i.e. if an hourly employee works 50 hours in a week, the first 40 hours would be paid by regular payment, remaining (50-40) 10 hours will be overtime payment that is 1.5 times the regular pay rate.
3. **CommissionEmployee**: Commission employees are paid a percentage of their total sales.
4. **BasePlusCommissionEmployee**: This group receives a base salary, additionally they also receive a percentage of their total sales.

Assume, further that, the management has decided to increase **Base-Plus-Commission-Employee’s basic pay by** 30% of their base salary. Their commission rate is also updated (Look inside the starter code for the amount). Complete the C# starter program, create appropriate derived classes and implement class methods/properties. The main program along with the base class and one derived class is already given to you. (**You do not need to modify the first two classes and the main method.**)

**Hints:**

* The base class **Employee** is an abstract type; your derived classes must override all abstract methods from the base class. Additionally, you must also override the ‘ToString()’ method of the **System.Object** class.
* **SalariedEmployee is already done for you as an example.**
* **CommissionEmployee** and **HourlyEmployee** are extended from the **Employee** class.
* Class **BasePlusCommissionEmployee**—which extends **CommissionEmployee**—represents the last employee type.



**Figure |** Employee hierarchy class diagram

**Evaluation**

This assignment has 3 main components, assigned marks as follows:

|  |  |
| --- | --- |
| Task | Marks |
| Written Questions | 30 |
| Code Quality (Uses of proper documentation comments, proper naming of variables, and code does not produce any compile-time errors) | 20 |
| Code Functionality (Every method is logically producing the correct output [not hard-coded constant value assigned], code is not using redundant computations and not using redundant variables). Standard OOP practices are followed, and code produces exact-identical input/output. | 50 |
| Total | 100 |

Marks for written questions are indicated beside each question.

To get full marks in coding, code must produce **identical input/output**. Your code must produce the exact correct output and must have the **correct logic to match the provided input/output**. Your code must be readable, well-documented, and well-designed according to standard OOP practices.