

Exam: Closed Internet, Open Book, Open Notes, Open Past Assignments

Part 1 (20 points; Approximate time: 15 minutes)

Add the Employee class to the project as follows:

- (a) Create 3 instance variables and 3 corresponding properties as follows. Each property will have nothing more than simple get / set accessors:
 - Instance variable name (string) and public property Name
 - Instance variable department (string) and public property Department
 - Instance variable salary (decimal) and public property Salary
- (b) Create 2 constructors
 - A default constructor that just has { } as a method body.
 - `public Employee(string name, string department, decimal salary)` – that initializes the 3 properties.
- (c) Create a method that returns a formatted string by inserting a tab character between each field. The form below shows the output created by this method. Use the following method header
 - `public virtual string getDisplayText() // format salary as currency!`
- (d) Create an abstract method that returns bonus as follows:

```
public abstract decimal getBonus();
```

Part 2 (20 points; Approximate time: 25 minutes)

Using Employee as the base class, add a class ManagementEmp class that inherits from the Employee class.

- (a) It should use the base class (Employee class) constructors and other methods as much as possible.
- (b) Add a variable as follows:

```
Instance variable years (decimal) and public property Years
// Years refers to the number of years of service for this employee
```
- (c) It should have a constructor that takes four input parameters: name, department, salary, years
- (d) Add a getPowerValue method to compute a measure of the power and value of the management employee. Use the following method header:

```
public decimal getPowerValue()
```

This method should return a number that is equal to $1000 * \text{years} + \text{salary}$
- (e) Write the getBonus() method: Bonus must be computed as 10% of the salary.
- (f) Override the getDisplayText() method to include the years. Must use the base class method.

Part 3 (20 points; Approximate time: 20 minutes)

Using Employee as the base class, add a class ClericalEmp class that inherits from the Employee class.

- (a) It should use the base class (Employee class) constructors and other methods as much as possible.
- (b) Add a variable as follows:

```
Instance variable sickleave (int) and public property Sickleave
// Sickleave refers to the amount of sick leave this employee has
```
- (c) It should have a constructor that takes four input parameters: name, department, salary, sickleave
- (d) Add a getTotalSickLeaveValue method to compute the total value of the sick leave. Use the following method header:

```
public decimal getTotalSickLeaveValue()
```

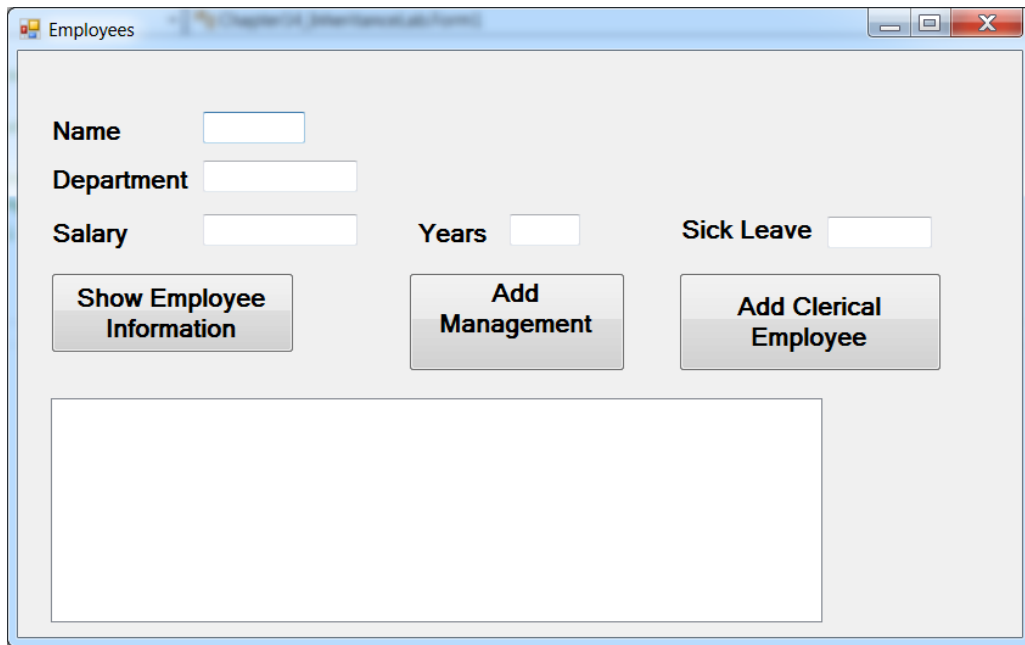
This method should return a number that is equal to salary * sickleave /2000;

- (e) Write the getBonus() method: Bonus must be computed as 5% of the salary.
- (f) Override the getDisplayText() method to include the sickleave. Must use the base class method.

Part 4 (40 points; Approximate time: 40 minutes)

Create the following form that uses the Employee class as follows

- 1) The form will have an instance variable “employees” that is a List of Employees. The user interface should be as follows:



- 2) Your code needs to add a management employee to the list when the user clicks “Add Management Employee” button.
- 3) Your code needs to add a clerical employee to the list when the user clicks “Add Clerical Employee” button.
- 4) When the user clicks the button, “Show Employee Information”, you code should do the following:
 - Show each employee’s details in the list box using the get display text method
 - The total salaries of all employees in the list box (formatted as currency).
 - The total of Power Values of all Management Employees in the list box.
 - The total of Sick leave Values of all Clerical Employees in the list box (formatted as currency).
- 5) Add a label to the form that shows the total number of employees. Update the label with the total number of employees in the employees list.
- 6) Use validation for input values and include try catch blocks.