

***C# BASICS***

**Training Assignments**

| **Document Code** | **25e-BM/HR/HDCV/FSOFT** |
| --- | --- |
| **Version** | **1.1** |
| **Effective Date** | **20/11/2012** |

**Hanoi, 06/2019**

**RECORD OF CHANGES**

| **No** | **Effective Date** | **Change Description** | **Reason** | **Reviewer** | **Approver** |
| --- | --- | --- | --- | --- | --- |
|  | 01/Oct/2018 | Create new | Draft |  |  |
|  | 01/Jun/2019 | Update template | Fsoft template | DieuNT1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Contents**

[Day 3: Assignment 10: Basic Project Practice 4](#_heading=h.1fob9te)

[Objectives: 4](#_heading=h.3znysh7)

[Business needs: 4](#_heading=h.2et92p0)

[Working requirements: 4](#_heading=h.tyjcwt)

[Technologies: 4](#_heading=h.3dy6vkm)

[Store Data: 4](#_heading=h.4d34og8)

[Functional/User Interface Requirement: 5](#_heading=h.2s8eyo1)

|  | **CODE: Net.M.A010**  **TYPE: MEDIUM**  **LOC: 190**  **DURATION: 180 MINUTES** |
| --- | --- |

# Day 3: Assignment 10: Basic Project Practice

**Objectives:**

* Understand and practice with Classes, Objects, Inheritance, Encapsulation.
* Understand and practice with Polymorphism, Abstraction.
* Understanding the main difference between method overloading and overriding, between abstract class and interface.

**Business needs:**

Create a Window Console application bases on OOP, Exception Handing, IO, Collection to manage Human Resource (HR Management System). This assignment will cover all part of C# Basic.

**Working requirements:**

* Working environment: Visual Studio 2013 or higher.
* Delivery: Source code packaged in a compress archive.

**Technologies:**

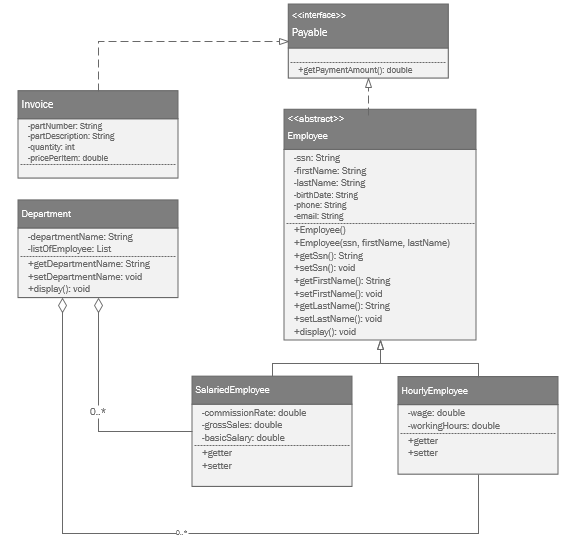
The product implements one or more technology:

* OOP
* Exception Handing
* Collection
* Control of Flows

**Store Data:**

* Data stored in collection classes.

For the class hierarchy is as follows, the trainee let's create the C# classes install this class diagram to be able to relationship between it.



* Employee is an abstract superclass and has six fields: *SSN*, *FirstName*, *LastName, BirthDate, Phone, Email*;
* A Salaried Employee is paid annually. Salaried employees are usually supervisory, managerial, or professional employees who work on an annual basis and are not paid an hourly rate. SalariedEmployee is a concrete class that is a subclass of Employee and adds 3 fields: CommisstionRate, GrossSales, BasicSalary;
* HourlyEmployee: Unlike a salaried employee who is paid a flat salary regardless of how many hours worked during a work month, an hourly employee is paid an hourly wage for each hour worked. This is a concrete class that is a subclass of Employee and adds two fields: *Rate*, *WorkingHours*;
* Department: Each department will have a list of employee (contains: SalariedEmployee and HourlyEmployee);

*Noting that:* each abstract class and concrete class has a constructor and methods for getting and setting its fields (getters and setters) and a ToString() method.

Program requirements must validate the properties:

* BirthDate : correct date format (dd/MM/yyyy);
* Phone: minimum 7 positive integers;
* Email: correct email format.

## Functional/User Interface Requirement:

The program has a screen console for UI

The **DepartmentManage** main method**:**

The main method in DepartmentManageclass uses the Employee class and its subclasses, the main screen allows selecting the functions for:

**1.** **Input data from the keyboard**: create an employee list of all types as mentioned above and belong to several departments.

**2.** **Display employees**: displays information about each object polymorphically. The objects are stored in an Employee array.

**3. Classify employees**: the last for loop illustrates how to find out the specific class for each object.

**4.** **Employee Search**:

* Search for employees by enter the name of a specific department, display the list of employees by type to the screen;
* Enter the employee's name, display detailed information about the employee

**5.** **Report**: display the list of departments and the number of employees for each.