# Working with NULL’s

Assume Employee class contains fields as below with its respective datatype. First Name and Last Name is mandatory where as other is not mandatory for the user to enter a value.

|  |  |  |
| --- | --- | --- |
| **FieldName** | **Type** | **IsMandatory** |
| FirstName | String | Yes |
| LastName | String | Yes |
| MiddleName | String | No |
| DateOfBirth | DateTime | Yes |

1. Write a program to accept the details and store them in Employee object. System should prompt user to enter employee details as below

Enter First Name

Enter Last Name

Enter Middle Name

Enter Date of birth

Enter Month of birth

Enter Year of birth

Once details of DOB is obtained, create an employee object and store the values obtained. Create DateTime object from the date,month and year information obtained and store them in employee object.

1. Introduce logic to check if an empty string is provided to MiddleName as an input then MiddleName field in the employee object should contain NULL.
2. Once all information is obtained and stored in the object, write code to display the information stored in the object as below. If middle name information is not provided(NULL) then print “-”

Employee First Name:

Employee Last Name:

Employee Middle Name:

Employee Date of Birth:

Concept learnt: String empty check, String null check, Using ??

1. In continuation with the above Employee object,
   1. Make DateOfBirth field also as optional i.e. DateOfBirth field should be able to hold null when no data is provided by the user.
   2. Display the employee object details in the below format. When middle name or last name is NULL then print “-”

Concept learnt: Nullable<T>, Using ??

1. In continuation with the above Employee object,
   1. Write code to find the age of the employee as of today and display it with other information as you print other information as below. If the DateOfBirth is not provided then print “-” against age

Employee First Name:

Employee Last Name:

Employee Middle Name:

Employee Date of Birth:

Employee Age:

Concept learnt: Using ?, Using ??

1. In continuation with the first example, refactor lines of code that prints to console in to separate method if not already done. Refactor it as below e.g.

Public void DisplayEmployeeInformation(Employee empObj)

{

// Lines of code that displays the information

}

1. After refactoring it into a method there is a possibility that any method could call DisplayEmployeeInformation and pass NULL as empObj e.g. DisplayEmployeeInformation(null). In such cases challenge is to make sure that the code written to display information does not break. How do you achieve it?

Concept learnt: Using ?.

1. Using Employee class as a base, create a List of 5 employees. Write program to search the given name in the list and print details on the console. When the given name to search does not match any name in the list, display information as below

Employee First Name: -

Employee Last Name: -

Employee Middle Name: -

Employee Date of Birth: -

Employee Age: -

You can hardcode the 5 employee details in the code. You have to request user to enter the name to search in the list, search and display information accordingly.

Hint: Use NULL object pattern to pass NullEmployee object to method DisplayEmployeeInformation

Concept learnt: Null object pattern