

***C# Basics***

**Lab Guides**

|  |  |
| --- | --- |
| 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 |  |  |
| 3 | 15/Apr/2019 | Review content | Review | TuTB |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Lab 11: Work with Abstract Class 4](#_Toc17709319)

[Objectives: 4](#_Toc17709320)

[Prerequisites: 4](#_Toc17709321)

[Problem Description: 4](#_Toc17709322)

[Guidelines: 4](#_Toc17709323)

[Step 1: Create project named **Abstract** in Visual Studio 4](#_Toc17709324)

[Step 2: Add class named **FlyingObject** then declare property and abstract method: 4](#_Toc17709325)

[Step 3: Create Bird class inherit from FlyingObject class 4](#_Toc17709326)

[Step 4: Create AirCraft class inherit from FlyingObject class 5](#_Toc17709327)

[Step 5: In Program.cs file, write code to initial objects in various ways 5](#_Toc17709328)

|  |  |
| --- | --- |
|  | **CODE: Net.S.L011**  **TYPE: SHORT**  **LOC: 100**  **DURATION: 30 MINUTES** |

# Lab 11: Work with Abstract Class

Objectives:

* Understand use Abtract class in C #.

Prerequisites:

* Download and installs Visual Studio (included .net Framework)

Problem Description:

* Create an FlyingObject class as an abstract class
* Create a Bird class inherit from FlyingObject class
* Create an AirCraft class inherit from FlyingObject class

Guidelines:

### Step 1: Create project named **Abstract** in Visual Studio

### Step 2: Add class named **FlyingObject** then declare property and abstract method:

abstract class FlyingObject

{

public string Name { get; set; }

public abstract void Fly();

}

### Step 3: Create Bird class inherit from FlyingObject class

class Bird : FlyingObject

{

public override void Fly()

{

Console.WriteLine("Bird is flying!");

}

public void Eat()

{

Console.WriteLine("Bird is eating!");

}

}

### Step 4: Create AirCraft class inherit from FlyingObject class

class AirCraft: FlyingObject

{

public override void Fly()

{

Console.WriteLine("Air Craft is flying!");

}

public void PourFuel()

{

Console.WriteLine("Air Craft is pouring fuel!");

}

}

### Step 5: In Program.cs file, write code to initial objects in various ways

static void Main(string[] args)

{

//// Cannot create object from abstract class

//FlyingObject flyingObject = new FlyingObject();

//// Create bird object

Bird bird = new Bird();

bird.Fly();

bird.Eat();

//// Create air craft object

AirCraft airCraft = new AirCraft();

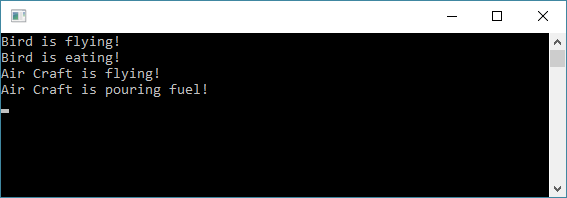
airCraft.Fly();

airCraft.PourFuel();

Console.ReadKey();

}

Outputs

****