**C#.NET Exercise 3**

**Contents**

[Lab Objective 3](#_Toc415141344)

[EXERCISE1: Creation of class to find Area and Circumfarences of circle and rectangle 3](#_Toc415141345)

[Task1: Creating a New Solution “Lab3” 3](#_Toc415141346)

[Task2: Creating new project “Exercise1” to “Lab3” Solution 3](#_Toc415141347)

[Task3: Create a new class Name it as Shape1 4](#_Toc415141348)

[Task4: Object Creation in Main Method 4](#_Toc415141349)

[EXERCISE2: Task Related to Inheritance and properties 6](#_Toc415141350)

[Task1: Creating new project “Exercise2” to “Lab3” Solution 6](#_Toc415141351)

[Task2: Create a New Class Name “ House” 6](#_Toc415141352)

[EXERCISE3: Task Related to to create a small domestic accounting system. 9](#_Toc415141353)

[Task1: Creating new project “Exercise3” to “Lab3” Solution 9](#_Toc415141354)

[Task2: Create a New Class Name “ HouseholdAccounts” 9](#_Toc415141355)

[EXERCISE4: Task Related to inheritance 10](#_Toc415141356)

[Task1: Creating new project “Exercise4” to “Lab3” Solution 10](#_Toc415141357)

[Task2: Create a Person class and inherit it to Student and Teacher classes 10](#_Toc415141358)

[EXERCISE5: Task Related to find magnitude using with real and complex numbers 12](#_Toc415141359)

[Task1: Creating new project “Exercise5” to “Lab2” Solution 12](#_Toc415141360)

[Task2: C# code to create a “ComplexNumber” class and “ ComplexTest” Class. 12](#_Toc415141361)

[Lab Summary 14](#_Toc415141362)

[Appendix: Code Samples (used if not running sequentially through the lab) 14](#_Toc415141363)

[EXERCISE 1: 14](#_Toc415141364)

[EXERCISE 2: 16](#_Toc415141365)

[EXERCISE 3: 18](#_Toc415141366)

[EXERCISE 4: 22](#_Toc415141367)

[EXERCISE 5: 23](#_Toc415141368)

**Lab Objective**

Estimated time to complete this lab: **60 minutes**

The objective of this lab is providing a clear understanding basics of c# programming and logical think of programming. In this lab you can understand of control-break statements clearly.

The Lab is structured to 4 Exercises. Each exercise contains several tasks. Each exercise presents different code.

In this lab you preformed the following exercises:

|  |
| --- |
| * Creation of class to find Area and Circumference of circle and rectangle * Task Related to Inheritance and properties * Task Related to create a small domestic accounting system. * Task Related to inheritance * Task Related to find magnitude using with real and complex numbers |

**EXERCISE1: Creation of class to find Area and Circumference of circle and rectangle**

create threeclass shape and overrides the methods Area() and Circumference() that are declared as abstract in Shape class and as Shape class contains abstract methods it is declared as abstract class.

**Task1: Creating a New Solution “Lab3”**

1. Click the **Start | Programs | Microsoft Visual Studio 2017 | Visual Studio 2017**.
2. On the **File** menu, point to **New,** and click **Project.**
3. In the **New Project** dialog box select the **other** project type.
4. Select the **Visual Studio Solution** template.
5. Provide a name for the new solution by entering “Lab3” in the **Name** box.
6. Click **OK**.

**Task2: Creating new project “Exercise1” to “Lab3” Solution**

1. Right Click on “ Lab3” Solution
2. On list choose “Add” and click on “New Project”
3. In the **New Project** dialog box select the **Visual C# Windows** project type.
4. Select the **Console Application** template.
5. Provide a name for the new project by entering “Exercise1” in the **Name** box.
6. Click OK.
7. You can able to see below structure of your project

**Task3: Create a new class Name it as Shape1**

1. In the **Solution Explorer**, double-click **Program.cs** to open the source code in the Visual C# code editor.
2. **Add new abstract class “Shape1” to “Exercise1” Namespace and create abstract methods like area and circumference.**

namespace Exercise1

{

abstract class Shape1

{

protected float R, L, B;

//Abstract methods can have only declarations

public abstract float Area();

public abstract float Circumference();

}

1. Create 2 classes like “Rectangle” and “Circle”. In “Rectangle” class get the Length and breadth values from the user and override area and circumferences methods to return area and circumferences of rectangle.
2. In “Circle” class create one method to ask user to enter Radius and override the methods like “area” and “circumferences” which are declared in “shape” abstract class. Implement those methods to return area and circumferences of circle.

**Task4: Object Creation in Main Method**

1. Create one method to print area and circumferences of shape looks below

public static void Calculate(Shape1 S)

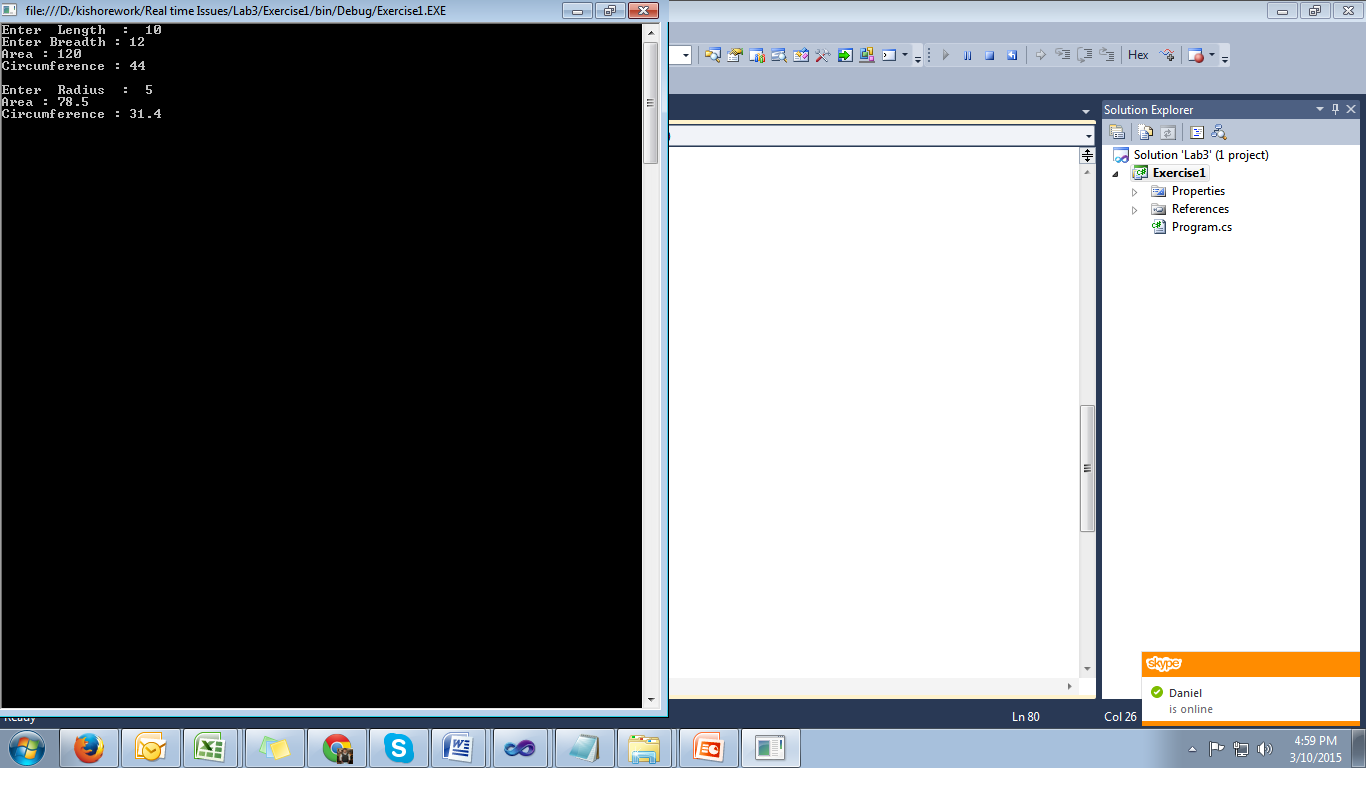
{

Console.WriteLine("Area : {0}", S.Area());

Console.WriteLine("Circumference : {0}", S.Circumference());

}

1. Create an object for Rectangle class And call the methods to enter length, breadth and to print area and circumferences of rectangle.
2. Create an object for circle class and call method to enter radius and also call methods like area and circumferences to return are and circumferences of circle.
3. Right click on your project “Exercise1” and click on Build (or) Press Ctrl+Shift+B to build the solution.
4. Press Ctrl+F5 to build and run the application. The program now compiles without any errors. Press any key to terminate the application.
5. The below pic shows you output window.



**EXERCISE2: Task Related to Inheritance and properties**

Create a class "House", with an attribute "area", a constructor that sets its value and a method "ShowData" to display "I am a house, my area is 200 m2" (instead of 200, it will show the real surface). Include getters an setters for the area, too.   
  
The "House" will contain a door. Each door will have an attribute "color" (a string), and a method "ShowData" wich will display "I am a door, my color is brown" (or whatever color it really is). Include a getter and a setter. Also, create a "GetDoor" in the house.   
  
A "SmallApartment" is a subclass of House, with a preset area of 50 m2.   
  
Also create a class Person, with a name (string). Each person will have a house. The method "ShowData" for a person will display his/her name, show the data of his/her house and the data of the door of that house.   
  
Write a Main to create a SmallApartment, a person to live in it, and to show the data of the person.

**Task1: Creating new project “Exercise2” to “Lab3” Solution**

1. Right Click on “Lab3” Solution
2. On list choose “Add” and click on “New Project”
3. In the **New Project** dialog box select the **Visual C# Windows** project type.
4. Select the **Console Application** template.
5. Provide a name for the new project by entering “Exercise2” in the **Name** box.
6. Click OK.

**Task2: Create a New Class Name “ House”**

1. Create a class name with “House” and declare variables like “area” and “door” which is from Door class. And also create a constructor to assign area value. Looks like below

class House

{

protected int area;

protected Door door;

public House(int area)

{

this.area = area;

door = new Door();

}

}

class Door

{

}

1. Create a properties for “House” class like “Area” and “Door” looks like below

public int Area

{

get { return area; }

set { area = value; }

}

public Door Door

{

get { return door; }

set { door = value; }

}

public virtual void ShowData()

{

Console.WriteLine("I am a house, my area is {0} m2.", area);

}

1. In “Door” class you can create afield like “Color”. And assign the color value in constructor and create properties also for color. Looks below

protected string color;

public Door()

{

color = "Brown";

}

public Door(string color)

{

this.color = color;

}

public string Color

{

get { return color; }

set { color = value; }

}

public void ShowData()

{

Console.WriteLine("I am a door, my color is {0}.", color);

}

1. Create a class “SmallApartment” and inherit it from “House” class. Create a constructor for this class and override “ShowData” method by passing some data looks like below.

class SmallApartment : House

{

public SmallApartment()

: base(50)

{

}

public override void ShowData()

{

Console.WriteLine("I am an apartment, my area is " +

area + " m2");

}

}

1. Create a “Person” class and declare variables like name and house. And also create constructor for this class and create properties for Name and House.[ here House related to “House” class] and also create a “ShowData” method to show your name.

class Person

{

protected string name;

protected House house;

public Person()

{

name = "Juan";

house = new House(150);

}

public Person(string name, House house)

{

this.name = name;

this.house = house;

}

public string Name

{

get { return name; }

set { name = value; }

}

public House House

{

get { return house; }

set { house = value; }

}

public void ShowData()

{

Console.WriteLine("My name is {0}.", name);

house.ShowData();

house.Door.ShowData();

}

}

1. Finally create a main method to test it, create main class “TestHouse”

class TestHouse

{

static void Main()

{

bool debug = true;

SmallApartment mySmallApartament = new SmallApartment();

Person myPerson = new Person();

myPerson.Name = "Kishore";

myPerson.House = mySmallApartament;

myPerson.ShowData();

if (debug)

Console.ReadLine();

}

}

**EXERCISE3: Task Related to create a small domestic accounting system.**

Create a program in C # that can store up to 10000 costs and revenues, to create a small domestic accounting system. For each expense (or income), should be allowed to save the following information:   
  
• Date (8 characters: YYYYMMDD format)   
• Description of expenditure or revenue   
• Category   
• Amount (if positive income, negative if an expense)   
  
The program should allow the user to perform the following operations:   
1 - Add a new expense (the date should "look right": day 01 to 31 months from 01 to 12 years between 1000 and 3000). The description must not be empty. Needless to validate the other data.   
  
2 - Show all expenses of a certain category (eg, "studies") between two certain dates (eg between "20110101" and "20111231"). Number is displayed, date (format DD / MM / YYYY), description, category in parentheses, and amount to two decimal places, all in the same line, separated by hyphens. At the end of all data show the total amount of data displayed.   
  
3 - Search costs containing a certain text (in the description or category without distinguishing case sensitive). Number is displayed, the date and description (the description is displayed in the sixth truncated blank, if any spaces six or more).   
  
4 - Modify a tab (tab number prompt the user, it will show the previous value of each field and press Enter to not be able to modify any of the data). Should be advised (but not re-order) if the user enters a wrong card number. Needless to validate any data.   
  
5 - Delete some data, from the number that you enter. Should be advised (but not re-order) if you enter an incorrect number. It should show the card to be clear and prompt prior to deletion.   
  
6 - Sort data alphabetically, by date and (if matched) description.   
  
7 - Normalize descriptions: remove trailing spaces, spaces and mirror sites. If a description is all uppercase, will be converted to lowercase (except for the first letter, kept in uppercase).

**Task1: Creating new project “Exercise3” to “Lab3” Solution**

1. Right Click on “Lab3” Solution
2. On list choose “Add” and click on “New Project”
3. In the **New Project** dialog box select the **Visual C# Windows** project type.
4. Select the **Console Application** template.
5. Provide a name for the new project by entering “Exercise3” in the **Name** box.
6. Click OK.

**Task2: Create a New Class Name “ HouseholdAccounts”**

1. Create a code to satisfy the requirement which is given in question.

**EXERCISE4: Task Related to inheritance**

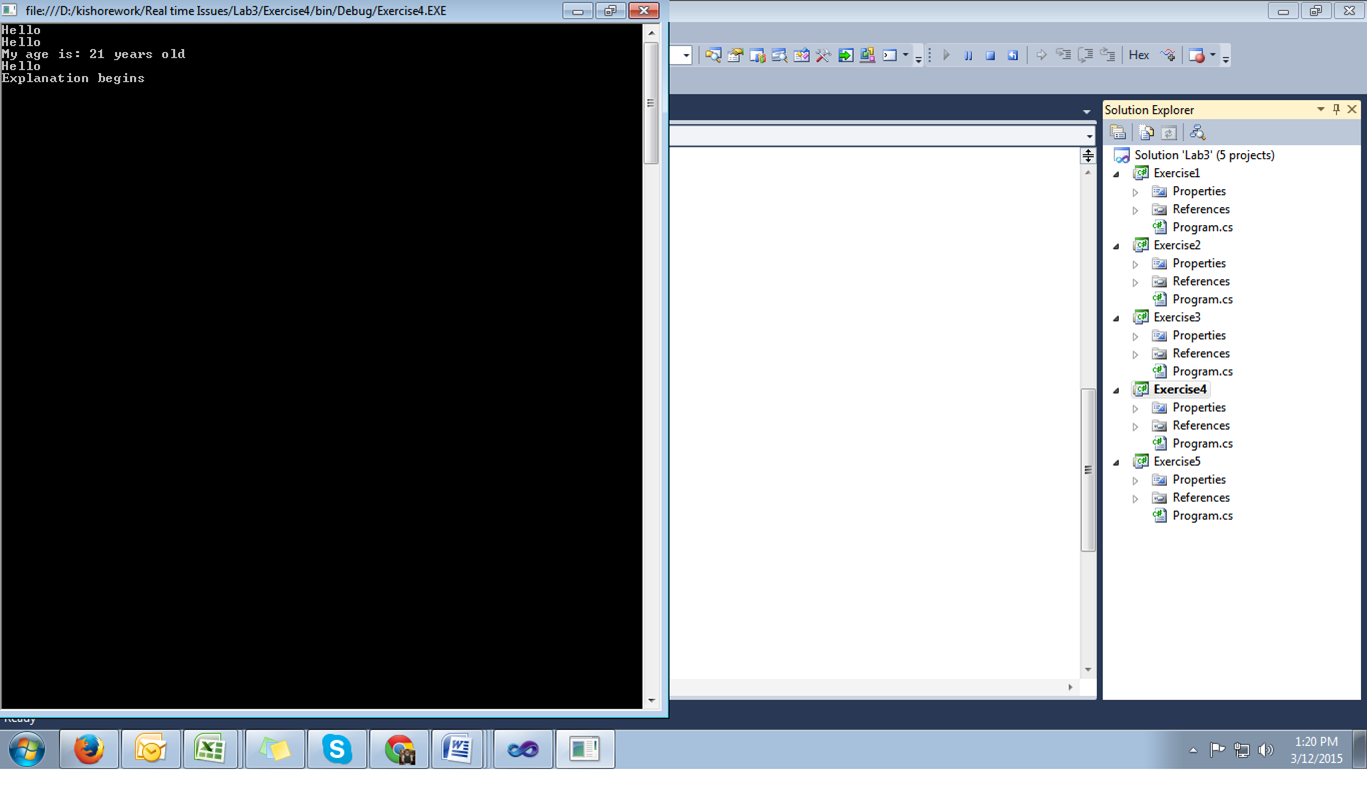
Create a class "Student" and another class "Teacher", both descendants of "Person".   
  
The class "Student" will have a public method "GoToClasses", which will write on screen "I’m going to class."   
  
The class "Teacher" will have a public method "Explain", which will show on screen "Explanation begins". Also, it will have a private attribute "subject", a string.   
  
The class Person must have a method "SetAge (int n)" which will indicate the value of their age (eg, 20 years old).   
  
The student will have a public method "ShowAge" which will write on the screen "My age is: 20 years old" (or the corresponding number).   
  
You must create another test class called "StudentAndTeacherTest" that will contain "Main" and:   
Create a Person and make it say hello   
Create a student, set his age to 21, tell him to Greet and display his age   
Create a teacher, 30 years old, ask him to say hello and then explain.

**Task1: Creating new project “Exercise4” to “Lab3” Solution**

1. Right Click on “Lab3” Solution
2. On list choose “Add” and click on “New Project”
3. In the **New Project** dialog box select the **Visual C# Windows** project type.
4. Select the **Console Application** template.
5. Provide a name for the new project by entering “Exercise4” in the **Name** box.
6. Click OK.

**Task2: Create a Person class and inherit it to Student and Teacher classes**

1. In the **Solution Explorer**, **Exercise4 Project**, double-click **Program.cs** to open the source code in the Visual C# code editor.
2. Write a code to meet all conditions mentioned in task.
3. The output should looks like below



**EXERCISE5: Task Related to find magnitude using with real and complex numbers**

A complex number has two parts: the real part and the imaginary part. In a number such as a+bi (2-3i, for example) the real part would be "a" (2) and the imaginary part would be "b" (-3).   
  
Create a class ComplexNumber with:   
A constructor to set the values for the real part and the imaginary part.   
Setters and getters for both.   
A method "ToString", which would return "(2,-3)"   
A method "GetMagnitude" to return the magnitude of the complex number (square root of a2+b2)   
A method "Add", to sum two complex numbers (the real part will be the sum of both real parts, and the imaginary part will be the sum of both imaginary parts)   
Create a test program, to try these capabilities.

**Task1: Creating new project “Exercise5” to “Lab2” Solution**

1. Right Click on “Lab3” Solution
2. On list choose “Add” and click on “New Project”
3. In the **New Project** dialog box select the **Visual C# Windows** project type.
4. Select the **Console Application** template.
5. Provide a name for the new project by entering “**Exercise5**” in the **Name** box.
6. Click OK.

**Task2: C# code to create a “ComplexNumber” class and “ ComplexTest” Class.**

1. In the **Solution Explorer**, **Exercise5 Project**, double-click **Program.cs** to open the source code in the Visual C# code editor.
2. Create a class name it as “ComplexNumber” and create a constructor and methods to meet the conditions given in question.
3. Create a class name “ComplexTest” , in this main method you can create an object for “ComplexNumber” and write code to print magnitude using methods, which is declared in “ComplexNumber” class.
4. Code looks like below

static void Main()

{

bool debug = false;

ComplexNumber number = new ComplexNumber(5, 2);

Console.WriteLine("Number is: " + number.ToString());

number.SetImaginary(-3);

Console.WriteLine("Number is: " + number.ToString());

Console.Write("Magnitude is: ");

Console.WriteLine(number.GetMagnitude());

ComplexNumber number2 = new ComplexNumber(-1, 1);

number.Add(number2);

Console.Write("After adding: ");

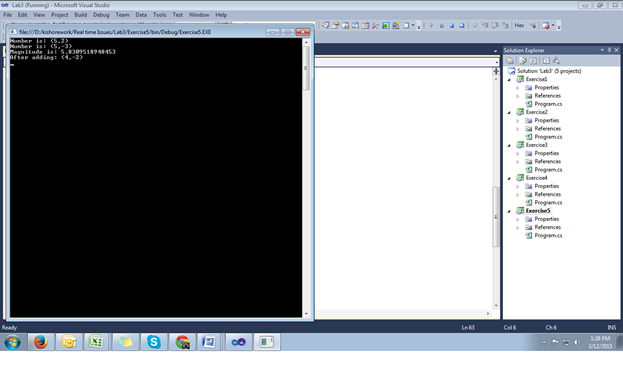
Console.WriteLine(number.ToString());

if (debug)

Console.ReadLine();

Console.ReadKey();

}

1. The output should looks like below 

**Lab Summary**