



# UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 2, Examination in Engineering, November 2016

Module Number: EE2201    Module Name: Object Oriented Programming

Part B

[2 hours]

[Answer all questions. Each question carries 5 marks.]

Q1. a) Explain *method overloading* in C#.

[1 mark]

b) The Listing 1 shows the code for Program class. What is the output of Usage() method?

Listing 1: Program class

```
public class Program
{
    static void F() { Console.WriteLine("F()"); }
    static void F(decimal x){
        Console.WriteLine("F(decimal)");
    }
    static void F(int x){ Console.WriteLine("F(int)"); }
    static void F(double x){ Console.WriteLine("F(double)"); }
    static void F<T>(T x){ Console.WriteLine("F<T>(T)"); }
    static void F(float y){ Console.WriteLine("F(float)"); }

    public static void Usage()
    {
        F();
        F(1);
        F(1.0);
        F(1d);
        F(1m);
        F("abc");
        F<int>(1);
        F(1f);
    }
}
```

[2 marks]

c) Briefly explain the differences of following items for C# language.

- i) structure and class
- ii) reference type and value type variables
- iii) private and public access modifiers
- iv) private and protected access modifiers

[2 marks]

Q2. a) Describe the following terms found in Object Oriented Programming.

- i) Encapsulation
- ii) Inheritance
- iii) Polymorphism

[1.5 marks]

b) You need to represent a point which has  $x$  and  $y$  coordinates in a C# program. Write code for Point class which has following requirements.

- i) Two auto-implemented properties that represents  $x$  and  $y$  coordinate.
- ii) A constructor to take two parameters for  $x$  and  $y$  coordinates.
- iii) Write `void Print()` method which prints  $x$  and  $y$  coordinates in the console as "Point  $x = 10.5$ ,  $y = 5.4$ ".
- iv) Overload **operator** `+` to add  $x$  coordinates and  $y$  coordinates of two points.
- v) Overload **operator** `*` to multiply  $x$  and  $y$  coordinates by a scalar number. (eg.  $2 * p$ , where  $p$  is a point.)

[3.5 marks]

Q3. a) Mention two differences between a concrete class and an abstract class in C# language.

[1 mark]

b) The Figure Q3.a shows the relationship among four classes. Classes Rectangle, Triangle and Circle derived from concrete Shape class. You need to call Draw() method using polymorphism. How you would implement in C#?

(You do not need to implement actual drawing inside the Draw() method. Print "drawing + shape name" when calling the method.)

[2 marks]

c) If the concrete Shape class in Figure Q3.a is replaced by abstract Shape class then how would you implement the scenario mentioned in section Q3b?

[2 marks]

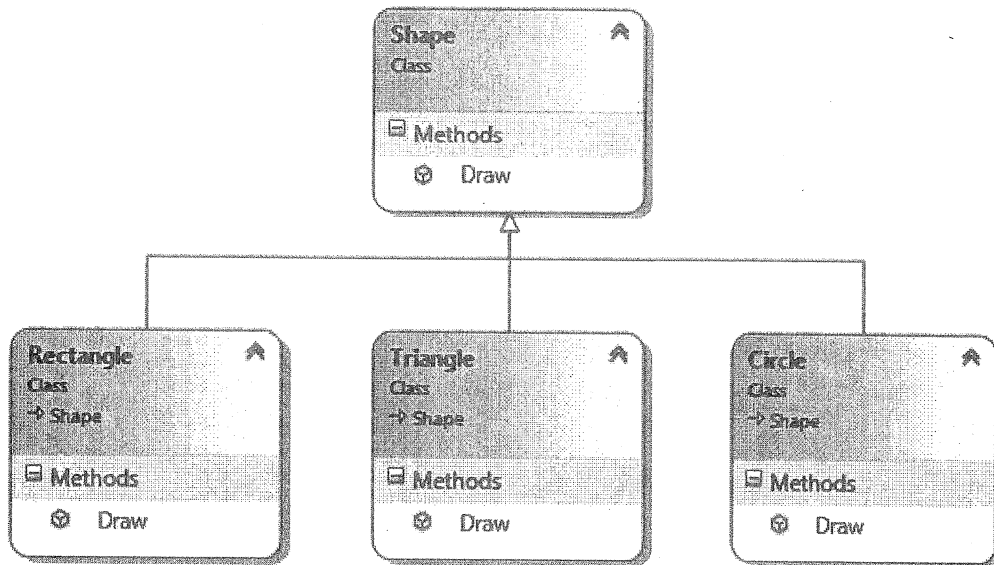


Figure Q3.a: Class diagram

Q4. The following program declares list of integers.

**Listing 2: Linq query test class**

```

using System;
using System.Collections.Generic;
using System.Linq;

class LinqTest
{
    static void Main(string[] args)
    {
        List<int> numberList = new List<int>{10, 23, 1, 11, 34, 34, 56,
            22, 64, 111, 26, 67, 59, 296, 5, 541, 78, 45, 14, 97 };

        //your code goes here

        Console.ReadKey(true);
    }
}
  
```

a) Write a Linq query to count number of even numbers.

[1 mark]

b) Write a Linq query to count numbers which are between 50 and 100.

[1 mark]

c) What is the content of variable list after executing the following Linq code?

```
var list= numberList.Where(p => p > 100)
                    .OrderByDescending( p => p)
                    .Select(p => new { num1 = p, num2 = p * p });
                    .ToList();
```

[1 mark]

d) Get a list of odd numbers in ascending order which are less than 100 using a Linq query.

[1 mark]

e) Print a list of numbers which are divisible by 3 in the Console.

[1 mark]

Q5. a) The following array class MyIntArray in Listings 3 can store set of integer numbers. Change the name of the class to MyGenericArray and modify the class to a generic class so that you can store any type of data in the class.

[2 marks]

**Listing 3: Int array class**

```
public class MyIntArray
{
    private int[] array;
    public MyIntArray(int size)
    {
        array = new int[size + 1];
    }

    public int getItem(int index)
    {
        return array[index];
    }

    public void setItem(int index, int value)
    {
        array[index] = value;
    }
}
```

- b) The code segment in Listing 4 shows a C# class. The expected output of the program is 25. It does not give the correct output. Explain the reason for the error and show how do you correct the program.

[1.5 marks]

**Listing 4: Method Example**

```
class Example
{
    static void MethodEx(int i)
    {
        i = i + 24;
    }

    static void Main()
    {
        int val = 1;
        MethodEx(val);
        Console.WriteLine(val);
    }
}
```

- c) The program in Listing 5 shows a program which read two numbers from the console. The program crashes sometimes. Explain a probable reason for the crash and explain how you would correct it.

[1.5 marks]

**Listing 5: Reading from console**

```
class Program
{
    static void Main(string[] args)
    {
        while (true)
        {
            Console.WriteLine("Enter Number 1");
            var num1 = Console.ReadLine();
            Console.WriteLine("Enter Number 2");
            var num2 = Console.ReadLine();

            var n1 = Convert.ToDouble(num1);
            var n2 = Convert.ToDouble(num2);
        }
    }
}
```