UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 2, Examination in Engineering, December 2015

Module Number: EE2201 **Module Name: Object Oriented Programming**

Part B

[1 hour and 45 minutes]

[Answer **all** questions]

Q1. a) Describe the following terms found in Object Oriented Programming.

i) Encapsulation

ii) Inheritance

iii) Polymorphism

[3 mark]

b) Explain the following access modifiers.

i) public

ii) protected

[2 mark]

c) The Listing 1 shows a partially implemented Circle class.

Listing 1: Circle class

```
class GradeBook
{
  public string CourseName { get; set; }

  public GradeBook(string name)
  {
    CourseName = name;
  }

  public void DisplayMessage()
  {
    Console.WriteLine("Welcome to the grade book " +
    "for {0}",CourseName);
  }
}
```

i) Include a second string auto-implemented property that represents the name of the course's instructor.

ii) Modify the constructor to specify two parameters-one for the course name and one for the instructor's name.

iii) Modify method `DisplayMessage()` such that it first outputs the welcome message and course name, then outputs "This course is presented by: ", followed by the instructor's name.

[3 marks]

Q2. a) Mention two differences between an abstract class and an interface in C# language.
[2 marks]

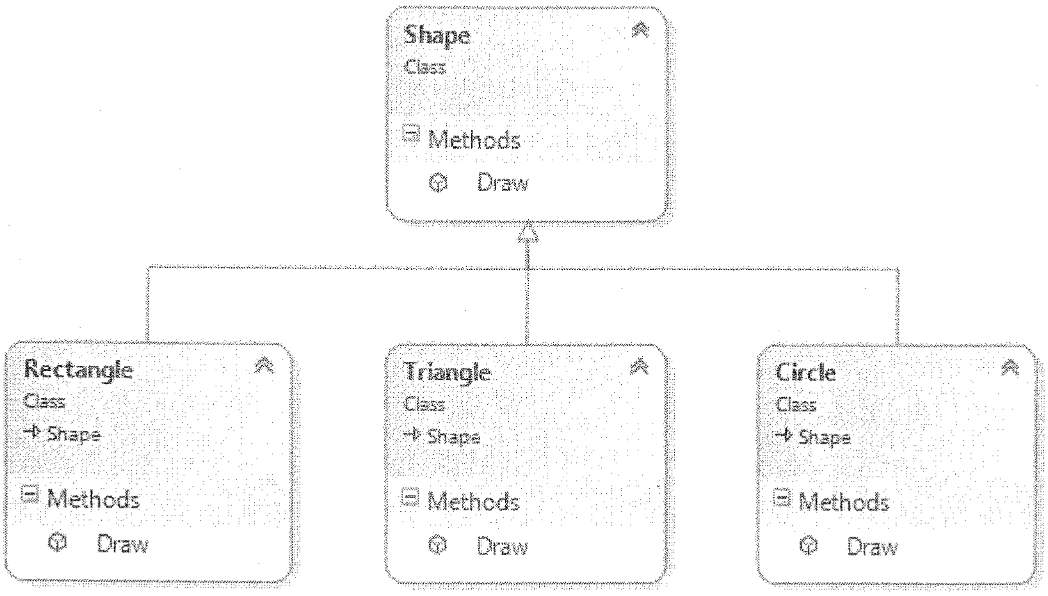

Figure Q2.a: Class diagram

b) The Figure Q2.a shows the relationship among four classes. Classes `Rectangle`, `Triangle` and `Circle` derived from `Shape` class. You need to call `Draw()` method using polymorphism. How you would implement in C#?

[2 marks]

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

c) If the Shape class in Figure Q2.a is replaced by abstract Shape class then how would you implement the scenario mentioned in section Q2b?

[2 marks]

d) If the shape class in Figure Q2.a is replaced by an interface, then how would you implement the scenario mentioned in section Q2b?

[2 marks]

Q3. Create a class called `Complex` for performing arithmetic with complex numbers. Complex numbers have the form $realPart + imaginaryPart * i$ where i is $\sqrt{-1}$. Use double precision floating-point variables to represent the private data of the class.

a) Provide a constructor that enables an object of this class to be initialized when real and imaginary parts are given.

[1 mark]

b) Provide a parameterless constructor with default values in case no initializers are provided.

[1 mark]

c) Provide public methods that perform the following operations:

i) Add two `Complex` numbers: The real parts are added together and the imaginary parts are added together.

[1.5 marks]

ii) Return a string representation of a `Complex` number in the form (a, b), where a is the real part and b is the imaginary part.

[1.5 marks]

d) Overload addition $(+)$ and multiplication $(*)$ operators in the `Complex` class.

[3 marks]

e) Write a class with `Main` method to test your `Complex` class.

[1 mark]