

UNIVERSITY OF RUHUNA

BACHELOR OF COMPUTER SCIENCE (GENERAL) DEGREE

LEVEL II (SEMESTER II) EXAMINATION – JANUARY 2018

COURSE UNIT: CSC 2213 – Rapid Application Development (Theory)

Duration: 2 hours

---

Answer **all** four questions

1)

- a) List two occasions where structures are used and two occasions where structures are not used in C#.
- b) Briefly explain the use of anonymous methods in C#. Use an appropriate example code to demonstrate the use of the concept.
- c) List three differences between delegates and events in C#.
- d) Write two main purposes of the Singleton design pattern.
- e) Write a C# code segment for the Singleton design pattern.

2)

- a) Draw a UML diagram to illustrate the Factory Method design pattern.
- b) Briefly explain the role of the Proxy design pattern.
- c) Briefly explain four types of the Proxy design pattern in relation to their usage in social networking related applications.
- d) Write four advantages of using Decorator design pattern.
- e) Use of specialized support tools is one major aspect of rapid application development. Write five such support tools along with their relevant application area.

3)

- a) Write the C# code for an interface (interface class) called *IComponent* that contains signature of a method returning a decimal value with an integer value in parameter list.
- b) Write the C# code for a class called *Component* which realizes the interface *IComponent* {in (3) (a) above}. The realized method should be implemented so that it returns factorial value of number given as argument to it.
- c) Use your answers (3) (a) and (3) (b) above and the Decorator design pattern to answer the questions (3) (c) (i) and (3) (c) (ii) given below.

- i. Write the C# code for a decorator class called *DecoratorA*. The realized method should return reciprocal of the factorial value of number given as argument.
- ii. Write a C# code for a decorator class called *DecoratorB*. Suppose the value given as argument to the realized method is  $n$ . Then the method should return value of the series given below.

$$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

- d) Write the C# code to evaluate the series given above in (3) (c) (ii) for  $n = 1000$  considering the classes you created above in (3) (a) to (3) (c).

4) Gmail is a free web based email service. With 2-Step Verification process for login, users can protect their account with their password and their phone. In 2-Step Verification process, first, the user has to provide correct username and password. Then the Google sends a secret code to the user's mobile phone which should be entered to the system in order to access the contents of the account. Considering the above description, use Proxy design pattern in C# to answer following questions.

- a) Write code to create an interface (interface class) which has only one method signature to return a string value.
- b) Write code to create a class called *Subject* which contains a single method to return the string "Welcome".
- c) Write code to create a proxy class called *Proxy2* which includes a method for authentication with a single parameter that is used for phone verification code. The return type of the authentication method must be void. If the authentication is successful then an object from the class *Subject* {in (4) (b) above} should be created.
- d) Write code to create another proxy class called *Proxy1* which includes a method for authentication with two parameters: *username* and *password*. The return type of the authentication method must be void. If the authentication is successful then an object from the class *Proxy2* {in (4) (c) above} should be created.

**Note:** Assume that two methods: the method *sendCode* that sends the verification code to phone, and the method *getCode* that returns the code entered by the user to the system have already been implemented for you. The class *Subject* {in (4) (b) above} should be accessed through the classes *Proxy1* and *Proxy2* sequentially.