
# Sri Lanka Institute of Information Technology

## B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 2, Semester 2 (2019)

## IT2020 – Software Engineering

Duration: 3 Hours

October 2019

Instructions to Candidates:

- This paper has five questions.
- Answer all questions in the booklet given.
- The total marks for the paper is 100.
- This paper contains 7 pages, including the cover page.
- Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.

## Question 1 — 20 Marks

a. Draw a sequence diagram for the following use case scenario "Add Items to Shopping Cart".

(20 marks)

| **Name:** Add Items to Shopping Cart |
|---|
| **Actor**: Customer |
| **Flow of Events:** |
| 1. Customer login to the system by providing username and password. |
| 2. Customer search for items by item names. |
| 3. Then the customer adds the selected items to shopping cart. |
| 4. If cart is not empty, the customer can remove items anytime. |
| 5. Customer can add any number of items to the cart. |
| 4. Once finished, the customer requests to process the order. |
| 5. Then system will generate and display the bill. |
| **Extensions** |
| 1-a. If login fails the customer will ask to login again. |

Hint: You may use two boundary, two control and three entity classes in your answer.

## Question 2 — 20 Marks

Given below is a detailed description of an online student registration system for an academic institute. Model a **physical diagram** according to the given description. (20 marks)

The system consists of Web server, Application server, Registration server and Student server. The Web server and Application server are deployed in two Oracle x86 server devices while Registration server and Student server are deployed in two Oracle SPARC servers. All servers installed Oracle Solaris 10 operation system.

Web server contains an application called Student_Registration while the Application server contains Student, Registration_system, Register and OWL_Registration_system applications. The OWL_Registration_system is an online web learning application. The Student application is a war file which uses the deployment specification in web.xml file. Web.xml file resides within the same Application server.

Registration server contains Registration_database component and Student server contains Student_Registration_System component. Student_Registration_System implements IStudent interface, which is used by OWL_Registration_System. Further, Registration_System communicates with the Registration_Database component through IRegistration interface, which

is implemented by Registration_Database. Registration_Database component uses Student_Registration application through IDatabase interface. Also Register component implements IRegister interface which uses by OWL_Registration_System.

Two Oracle x86 servers communicate with two Oracle SPARC servers over local area network. Web server and Application server communicates with each other over TCP/IP protocol.

## Question 3 — 20 Marks

a. How many minimum number of test cases are needed for full statement coverage of following code segment? Calculate percentages for each test case. (8 Marks)

```
import java.util.Scanner;

public class NestedIf {
        private static Scanner sc;

        public static void main(String[] args) {
                int age;
                sc = new Scanner(System.in);
                System.out.println(" Please Enter you Age: ");
                age = sc.nextInt();

                if (age < 18) {
                        System.out.println("You are Minor.");
                        System.out.println("You are Not Eligible to Work");
                }
                else  {
                        if (age >= 18 && age <= 60 )  {
                        System.out.println("You are Eligible to Work");
                        System.out.println("Please fill in your details and
                         apply");
                        }
                        else  {
                        System.out.println("You are too old to work as per the
                         Government rules");
                        System.out.println("Please Collect your pension!");
                        }
                }

        }
}
```

b. Consider the partial code segment given above.

i. Draw a control flow graph for the partial code segment given below. (4 marks)

ii. Calculate the minimum number of test cases required for a full branch coverage (Show the branch coverage as a percentage for each test case). (8 marks)

```
class Number {
    public static void main(String[] args) {

        Double n1, n2, n3, largestNumber;

        if (n1 >= n2) {
            if (n1 >= n3) {
                largestNumber = n1;
            } else {
                largestNumber = n3;
            }
        } else {
            if (n2 >= n3) {
                largestNumber = n2;
            } else {
                largestNumber = n3;
            }
        }

        System.out.println("Largest number is " + largestNumber);
    }
}
```

## Question 4 — 20 Marks

a. Consider the below given scenario of registering a new member in a web site.

A client needs to go through a registration process to get a membership of a web applications. As part of a successful registration process, need to complete a series of steps such as saving the registration details, creating a new member, processing payment and sending a welcome email. In a well-designed system different components are responsible for each of these operations. Assuming that you are the software engineer to design the registration process, answer the following questions.

Hint: Assume that this entire registration process requires Member_services, New_member, Payment and Email classes.

i. Suggest an appropriate design pattern for this scenario. (1 mark)
ii. Give an advantage of using the design pattern you suggested in part c) - i). (1 mark)
iii. Explain the design pattern you suggested for this scenario using a simple class diagram. (4 marks)

b. Consider the below given scenario of "Cozze Cakes" cake ordering system and answer the questions.

"Cozze Cakes" is popular for wide variety of cakes for any occasion. Customers can order cakes in different flavors such as vanilla, chocolate, Ribbon, Red velvet and Marble. Also they have many cake filling flavors such as Chocolate cream, Butter cream, Lemon, Caramel, Cream Cheese and Strawberry. Each cake flavor and filling flavor has its own price. Customers can order cakes with different combinations of filling flavors as they wish.

ii. Suggest an appropriate design pattern for this scenario. (1 mark)
iii. Write a reason for using the design pattern you suggested in part b) - i) for the scenario mentioned above. (1 mark)
iv. Explain the design pattern you suggested for this scenario using a simple class diagram. (4 marks)

c. Consider the partial code segment given below and answer the subsequent questions.

```
import java.util.Map;

interface StockBroker {
    void update(Map<String, Double> stockList);
}

import java.util.Iterator;
import java.util.Map;

public class StockBuyer implements StockBroker {
    public void update(Map<String, Double> stocks) {
        ..........
..........
    }
}
public class StockViewer implements StockBroker {
    public void update(Map<String, Double> stocks) {
        ..........
..........
    }
}
public abstract class AbstractStockMarket {
    ..........
    public void addStockBroker(StockBroker stockBroker) {
        stockBrokers.add(stockBroker);
    }
    public void notifyStockBroker(Map<String, Double> stockList) {
        for (StockBroker broker : stockBrokers) {
            broker.update(stockList);
        }
    }
    public abstract void addStock(String stockSymbol, Double price);
    public abstract void update(String stockSymbol, Double price);
}
public class StockMarket extends AbstractStockMarket {
    ..........
    public void addStock(String stockSymbol, Double price) {
        stockList.put(stockSymbol, price);
    }
    public void update(String stockSymbol, Double price) {
        stockList.put(stockSymbol, price);
        notifyStockBroker(stockList);
    }
}
public class Application {
    public static void main(String[] args) {
        AbstractStockMarket market = new StockMarket();
        StockBroker buyer = new StockBuyer();
        StockBroker viewer = new StockViewer();
        market.addStockBroker(buyer);
        ..........
    }
}
```

i. Identify the design pattern used in the code. (1 mark)

ii. What is the main purpose of using the design pattern you have mentioned above for this solution? (1 mark)

iii. Draw the class structure of the design pattern that you identified in part i with appropriate methods for the above scenario. (6 marks)

## Question 5 — 20 Marks

a. Draw a state machine diagram for the following scenario which describes the operation of a Virus Guard. (16 marks)

A Virus Guard is automatically loaded when the computer starts up and is initially in the checking state where it will do a self-scan and check if the virus definitions files are up to date. If there is an error in the self-scan, the virus guard changes its state to the critical state. In the critical state, the virus guard repair the virus definition files. Once finished, the virus guard move back to checking state. If the virus definitions are up to date the Virus Guard changes its state to the active state. If the virus definitions is over one month old it changes its state to the major updating state. If the virus definitions are less than one month old, it changes its state to the minor updating state. In both cases it downloads the updates from the virus guard website. In the major updating state it disallows installation of new software and restricts the access to the internet and networks. In the minor updating state software installations from the web are disallowed. Once the updates are carried out successfully, the virus guard changes its state to the active state. If a virus is found and is a non-critical virus, then the Virus Guard changes to the Virus Alert State. Here it will clean the viruses from the source. If a virus which is of high risk or if an unknown virus is detected the virus guard moves into the critical virus alert state. In this state the network connections and all pen drives are disabled. The details of the virus are sent to the virus guard website through a secure tunnel. Once the virus is removed the system it would move to the active state.

b. Explain the difference between distributed version control and centralized version control. (2 marks)

c. Explain incident management? (2 marks)

---------- END OF THE QUESTION PAPER ----------