



Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 2, Semester 1 (June Intake)

IT2020 – Software Engineering

Duration: 3 Hours

November 2023

Instructions to Candidates:

- ◆ This paper has 5 questions.
- ◆ Answer all questions in the booklet given.
- ◆ The total marks for the paper is 100.
- ◆ This paper contains 10 pages, including the cover page.
- ◆ Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.

Question 1**(25 marks)**

Analyze the partial requirements of the Full Bright Education enrollment process (FBE-EU) given below and draw a sequence diagram.

Note: You may use suitable boundary, control, and entity classes.

“Full Bright Education (FBE) “is a famous agent profile that supports students who seek foreign education in England. With the high demand of the current student community and strict rules and regulations in the UK, the enrollment process plays a vital part in the foreign education process. Therefore, the Team FBE needs a sophisticated system to process enrollments.

Any student can search for available universities and their enrollment requirements by giving their details such as age, current qualification, area of study, and study level. After displaying the requested details on the MainUI, the FBE system will store the given details in its Programs_Core database. Once stored, the system will analyze the current trends and advertise them automatically in the MainUI.

Students who seek enrollment for the very first time need to register with the FBE by providing a valid passport number and their full name through LoginUI. As a result of the registration, they will receive the UFE number. Please note that registration has already been modeled for reuse purposes. By providing details such as UFE number, area of study, and university, students can book a discussion with the agent profile via MainUI. Then the system will store the details in the Programs_Core database and will return a notification of the discussion. Then the system will check the counselor details in the counselor database and then assign counselors randomly based on the universities and counselors’ expertise areas (IT, business marketing, and so on). Then the assigned counselors will be notified to the student.

After the discussion process, the students must enter all the details, including the UFE number and request for the offer, through the system. Then the system will verify the offer details via the Programs_Core database. Please note that offer verification has already been modeled and is ready to use. Then the system sends the details to a third-party Uni portal called IMMI.gov.uk. Then, the IMMI controller will check and validate the details via IMMI

database. For successful offers, IMMI controller may send a successful notification through the system and sends the offer directly to the mainUI. In the same manner, they may send a regret notification for unsuccessful offers. Simultaneously, the IMMI controller will update the status via the Programs_Core database to notify the FBE of the status of the candidate.

Question 2 **(20 marks)**

Below is a detailed description of an application developed for a higher education platform called Millenium Distance Learning Platform (MDLP). Model a **physical diagram** according to the given description.

Note: you need to include the necessary operating systems in the appropriate places.

With the increasing demand for higher education after the AL examination, the Ministry of Education in partnership with the leading institute in Sri Lanka plans to propose a distance learning platform known as MDLP - Millenium Distance Learning Platform, which can be accessed over the web or mobile as per the need of the student.

When students try to connect with the MDLP via mobile, they must install MLDP_MobAPP on their phones. If they use any personal computer (Desktop, Laptop), they must use the MLDP web app via a web browser. The main MLDB Dashboard has multiple User Interfaces (UIs) to support Registration, Login, and Program sub-systems. It is in a main webserver which is installed inside a Dell PowerEdge R720 Server.

The first-time user can register with the MDLP platform which connects the user with the Main Server system through the iReg interface implemented by Registration UI. Already registered users can use the iLogin interface to connect with the Login component. After login in with the MDLP, the user can find details about the available programs, where he/she connects with the main server using iProgram interface realized by Program sub-system. The program subsystem implements iConnect interface to connect to Login and Registration subsystems.

The Lenovo ThinkSystem SR650 server runs the services related to the MDLP Distance learning platform. Where the application Server runs 3 sub-systems related to user

authentication, user profile management, and program enrollment. The user authentication sub-system has two components, the user verification component, and the user session component. When someone registers into the MDLP, the Registration subsystem connects with the user verification through the iVerify interface. A logged in user connects with the user profile management sub-system via iPM interfaces implemented by user profile management. A logged-in user can access the program details through the program component by connecting to the iProg interface implemented by the program enrollment sub-system.

The data sources are kept in a separate Data Center which runs in a IBM B2 Server. There, it has an instance of the database component residing in the Microsoft SQL Server environment, and the sub-systems user Authentication, User profile Management, and Program enrollment are connected to the Database component through the iData interface implemented by the Database component. Database.dll file which resides inside the same IBM B2 server manifests the database component.

Desktop and mobile users connect to the Dell PowerEdge server via TCP/IP protocol while Lenovo Think System connects to the Dell PowerEdge server via LAN. Lenovo Think System connects to the data center via WAN.

Question 3**(25 marks)****Part 01**

Based on the code given below, answer the questions at the end.

(15 marks)

```
#include <iostream>
using namespace std;
int main() {
    char choice;
    do {
        int packageNo;
        double totalDistance, totalAmount;
        cout << "Package No: " << endl;
        cout << "1. Comfort Journey" << endl;
        cout << "2. Budget cab Journey" << endl;
        cout << "Enter the package number: ";
        cin >> packageNo;
        switch (packageNo) {
            case 1:
                totalAmount = 150.0 + 175.0 * totalDistance;
                break;
            case 2:
                totalAmount = 100.0 * totalDistance;
                break;
            default:
                cout << "Invalid Package Number!!!" << endl;
                continue;
        }
        cout << "Total Distance (in kilometers): ";
        cin >> totalDistance;
        cout << "Total Amount: Rs. " << totalAmount << endl;
        cout << "Do you have more customers? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');
    cout << "Program terminated." << endl;
    return 0;
}
```

- a) Draw the control flow graph for the given code above. (05 marks)
- b) Identify the number of branches available in the given code. (01 mark)
- c) Calculate the branch coverage percentage, when you enter packageNo = 2 and choice = 'n' respectively. (03 marks)
- d) Calculate the branch coverage percentage when you enter packageNo = 6, packageNo = 1 and choice = 'n' respectively. (03 marks)

- e) Calculate the branch coverage percentage when you enter packageNo = 1 and choice = 'Y', packageNo = 6 and choice = 'y', packageNo = 2 and choice = 'n' respectively.
(03 marks)

Part 02

Based on the code given below, answer the questions at the end. (10 marks)

```
i = 3
while i < 6:
    print(i)
    i += 1
    if i > 3:
        print ("Three or above three")
    elif (i == 3):
        print("three")
    else:
        print ("Less than three")
        break
print ("end of loop")
```

- a) Calculate the statement coverage percentage of the code given above. (03 marks)
- b) Calculate the statement coverage percentage when i=7. (02 marks)
- c) Calculate the statement coverage percentage when i=2. (02 marks)
- d) Calculate the statement coverage percentage when i=1. (03 marks)

Question 4**(20 marks)****a)**

Imagine that you are developing a drawing application. This application allows users to draw different shapes, such as circles, squares, and triangles. Each shape has a number of attributes, such as its position, size, and color. To reduce the memory usage of the application, it was decided to use the Flyweight design pattern.

i) Identify the intrinsic and extrinsic information of the shape object. (02 marks)

ii) Draw the relevant class diagram that can be used to represent the given scenario. (03 marks)

b)

In a university system, when selecting students for the best performance award, they consider several criteria besides academic CGPA. These criteria include Achievements in Sports, Achievements in coding competitions, Publishing research papers, Engaging in CSR Projects, and any other approved extracurricular activities. Each criterion will add a score to the CGPA and the student with the highest score will get the award. Students with high CGPA of bachelor's and master's degrees are eligible for the award. Each student may have more than one criterion satisfied as well as same criteria can be satisfied more than once.

i) Recommend the most suitable design pattern for the given scenario above and justify your recommendation. (02 marks)

ii) Draw the class structure of the design pattern that you identified above in part i) with appropriate classes and methods for the above scenario. (05 marks)

c) Consider the code given below and answer the questions at the end.

```
public class AlarmTracker {  
    private final List<Devices> devices = new ArrayList<> () ;  
    private int SensorReading;  
  
    public int getTheft() {  
        return SensorReading;  
    }  
  
    public void setTheftSensor(int state) {  
        this.SensorReading = state;  
        alarmDevices();  
    }  
  
    public void addDevice(Devices device){  
        devices.add(device);  
    }  
  
    public void alarmDevices(){  
        for (Devices devices : devices) {  
            devices.Trigger();  
        }  
    }  
}  
-----  
public abstract class Devices {  
  
    protected AlarmTracker alarm_Tracker;  
    public abstract void Trigger();  
}  
-----  
public class MainMonitor extends Devices {  
  
    public MainMonitor(AlarmTracker alarm_track) {  
  
        this.alarm_Tracker = alarm_track;  
        this.alarm_Tracker.addDevice(this);  
    }  
}
```

```
@Override  
public void Trigger() {  
    System.out.println(this.alarm_Tracker.getTheft() ) ;  
}  
  
-----  
  
public class MobilePhone extends Devices{  
  
    public MobilePhone(AlarmTracker alarm_track) {  
  
        this.alarm_Tracker = alarm_track;  
        this.alarm_Tracker.addDevice(this);  
    }  
  
    @Override  
    public void Trigger() {  
        System.out.println(this.alarm_Tracker.getTheft() ) ;  
    }  
}  
  
-----  
  
public class AlarmBell extends Devices {  
  
    public AlarmBell(AlarmTracker alarm_track) {  
  
        this.alarm_Tracker = alarm_track;  
        this.alarm_Tracker.addDevice(this);  
    }  
  
    @Override  
    public void Trigger() {  
        System.out.println(this.alarm_Tracker.getTheft() ) ;  
        //Ring Bell  
    }  
}
```

```
public class AlarmSystem2 {  
  
    public static void main (String[] args) {  
  
        AlarmTracker alarm_Tracker = new AlarmTracker();  
  
        Devices dv1 = new MainMonitor(alarm_Tracker);  
        Devices dv2 = new MobilePhone(alarm_Tracker);  
        Devices dv3 = new AlarmBell(alarm_Tracker);  
  
        // Door open = 15 , Unauthorized access = 10  
        alarm_Tracker.setTheftSensor(15);  
  
        alarm_Tracker.setTheftSensor(10);  
    }  
}
```

- i) Identify the design pattern used in the code given above. (01 mark)
- ii) Identify the main purpose of using the design pattern you have mentioned above in part i) for this solution. (02 marks)
- iii) Draw the class structure of the design pattern that you identified in part (i) with appropriate methods to represent the above scenario. (05 marks)

Question 5 (10 marks)

- a) Explain three types of internal behaviors in a state machine diagram with suitable examples for each. (03 marks)
- b) Differentiate between the Synchronous and Asynchronous messages. (02 marks)
- c) Identify one advantage and one disadvantage of using commercial and open-source software. (02 marks)
- d) List the main steps in the incident management process. (03 marks)

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