

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Final
Semester: Summer 2023

Duration: 1 Hour 50 min
Full Marks: 40

CSE 470: Software Engineering

SET A

Name:

ID:

Section:

Q1. A publishing company has hired you as the lead developer for a project to create a web-based content management system (CMS). The CMS must support user administration, publishing, editing, and content generation. The business has established clear and comprehensive specifications up front, and they have a set deadline for the project.

- a. Which software development approach would you suggest for creating the web-based CMS given the nature of the project and its clear and set requirements? Justify your decision and list the main stages of the selected model. [3.5 marks] [CO1]
- b. Which architectural pattern would you pick when creating a web-based CMS to ensure effective concern separation, modularity, and code reuse? Describe why the selected architecture meets the needs of the project. [3.5 marks] [CO5]

Q2. In the city of Neo-Gotham, "The Neo-Gotham Chronicle", a renowned newspaper agency, has converted from printing media to digital journalism. With a number of services including interactive e-magazines, real-time podcasts, and even an augmented reality news portal, the Chronicle has a variety of readers. However, the preferences of Neo-Gotham's population are really diversified, from seniors who cherish printed papers to tech-savvy younger generation who are eager for the latest news on their cell-phones and neural-network devices.

To maintain their popularity, the Chronicle introduced "ChronoLink", an innovative news delivery system powered by AI. However, ChronoLink has faced its share of challenges, from content adaptation struggles to linguistic barriers. Fearing potential misuse of ChronoLink, the Chronicle's CEO has confidentially tasked you with addressing some issues.

- a)** Consider the following code and Answer the following questions: [3 marks] [CO3]
- (i)** Implement the `notifySubscribers` method in the `NewsItem` class to update each subscriber about the changed content.
 - ii) Implement the `addSubscriber` method to add a user to the subscribers list.
 - iii) Implement the `removeSubscriber` method to remove a user from the subscribers list.
- b)** Implement the `receive Notification` method in the `User` class. This method should process the received news content, translating it to the user's preferred language. Additionally, depending on the user's device type (Neural-Interface or Traditional), prefix the content appropriately before displaying. [2 marks] [CO3]
- i) For example, sample news content to a traditional consumer:
 - ii) "General: Bangladesh won the 3 match T20 series against India."

- c) Describe how the User and NewsItem classes interact to ensure that when content in a NewsItem is updated, all subscribers (users) receive the modified content in their preferred language and format, tailored to their device. [2 marks] [CO3]

News Item Class:

```
public class NewsItem {
    private String title;
    private String content;
    private List<User> subscribers; // Observers list

    // Constructors, getters, and setters (You need not
    write this section)

    public String getContent () { Return content; }

    public void updateContent (String newContent) {
        this.content = newContent;
        // TODO: Notify all subscribers of the content
        change
        notifySubscribers();
    }

    public void notifySubscribers() {
        // TODO: Update all subscribers with the new
        content
    }

    public void addSubscriber (User user) {
        // TODO: Add a user to the subscribers list
    }

    public void removeSubscriber (User user) {
        // TODO: Remove a user from the subscribers
        list
    }
}
```

User Class:

In Neo-Gotham, citizens use different devices to consume news. For example, tech-lovers might use a "Neural-Interface" device that prefixes content with "Neural", while traditionalists might use standard devices that prefix content with "General".

```
public class User {
    private String preferredLanguage;
    private String deviceType;

    // Constructors, getters, and setters (You need not
    write this section)

    public void receiveNotification (NewsItem item)
    {
        // TODO: Process and display the news
        content to the user based on their device type
        and language preference
    }
}
```

Q3. Consider the following code snippet.

- Determine the Specialization Index (SIX) for the T20 World Cup. [3]
- Illustrate the Control Flow Graph (CFG) for the `hostCountry()` method, and remember to label the relevant node numbers on your question paper. [4]
- Calculate the cyclomatic complexity of the control flow graph drawn for the `hostCountry()` method. [3]
- Identify all the independent paths from the CFG and show one test case of any independent path you identified. Assume any data you need to make that test case. [5]
- Write down the difference between white box and grey box testing [4]

```
public class CricketCup {
    public void startCup() {
        System.out.println("Cricket Cup has started.");
    }
    public void endCup() {
        System.out.println("Cricket Cup has ended.");
    }
    private void displayWinner() {
        System.out.println("Winner of the Cricket Cup is
not yet decided.");
    }
}

public class WorldCup extends CricketCup {
    private void displayWinner() {
        System.out.println("Winner of the World Cup is
India.");
    }

    public void awardMVP() {
        System.out.println("The Most Valuable Player
(MVP) of the World Cup is Virat Kohli.");
    }
}

public class T20WorldCup extends WorldCup {
    public void awardMVP() {
        System.out.println("The Most Valuable Player
(MVP) of the World Cup is Virat Kohli.");
    }
}
```

```
public void hostCountry() {
    boolean isT20WorldCupHosted = true; // You can
    set this value based on your condition.
    int year = 2023; // You can set the year based on
    your requirement

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++) {
                if (i == 2 && j == 2 && k == 2) {
                    System.out.println("The T20 World Cup
is hosted in Australia in " + year + ".");
                } else if (i == 1 && j == 1 && k == 1) {
                    System.out.println("The T20 World Cup
host country is yet to be decided.");
                }
            }
        }
    }

    if (isT20WorldCupHosted) {
        System.out.println("The T20 World Cup has a
host country.");
    } else {
        System.out.println("The T20 World Cup does
not have a host country yet.");
    }
}

public void awardManOfTheTournament() {
    System.out.println("The Man of the Tournament
for T20 World Cup is Glenn Maxwell.");
}
}
```


- Q4. Consider the following code snippet.
- Identify all the code smells present in the following mentioning their names. [3]
 - Refactor the source code to remove all the identified code smells. [4]
- NB: The "System.out.println" statements should not cause any smells.

```
interface AbstractNut {
    void serve();
    double getUnitPrice();
    void addHoney(double honeyQuantity);
}

abstract class DriedNut implements AbstractNut {
    private double quantity;
    public DriedNut(double quantity) {
        this.quantity = quantity;
    }
    double getQuantity() {
        return this.quantity;
    }
    double getPrice() {
        return this.quantity * this.getUnitPrice();
    }
    public void serve() {
        System.out.println("Packaging nut of " +
            this.quantity + " kg.");
    }
    public void addHoney(double honeyQuantity) {
        System.out.println("Adding " +
            honeyQuantity + " kg honey to nuts.");
    }
}
```

```
class Almond extends DriedNut {
    Almond(double quantity) {
        super(quantity);
    }
    @Override
    public double getUnitPrice() {
        // 10% tax and Tk 5 for transportation
        return 1000 + 1000*0.1 + 5;
    }
}

class Cashew extends DriedNut {
    Cashew(double quantity) {
        super(quantity);
    }
    @Override
    public double getUnitPrice() {
        // 11% tax and Tk 5 for transportation
        return 900 + 900*0.11 + 5;
    }
}
```

(P.T.O)

```

class MixedDriedNuts extends DriedNut {
    double q1, q2;
    DriedNut a, c;

    public MixedDriedNuts(double quantity)
    {
        super(quantity);
        this.q1 = quantity*0.4;
        this.q2 = quantity*0.6;
        // Both this.q1 and this.q2 cannot be empty
        this.a = new Cashew(this.q1);
        this.c = new Almond(this.q2);
    }

    @Override
    public void serve() {
        System.out.println("Packaging
        MixedDriedNuts of " +
        this.getQuantity() + " kg.");
        this.a.serve();
        this.c.serve();
        // Mixing nuts
        System.out.println("Stir nuts.");
        System.out.println("Add spice.");
        System.out.println("Stir nuts
        again.");
    }

    @Override
    public double getUnitPrice() {
        // Tk 10 is mixing price
        return a.getUnitPrice()*0.4 +
        c.getUnitPrice()*0.6 + 10;
    }
}

```

```

class HoneyNut extends MixedDriedNuts {
    double q3; // Honey amount
    public HoneyNut(double quantity, double honey) {
        super(quantity);
        this.q3 = honey;
    }
    @Override
    public double getUnitPrice() {
        // Honey price Tk 300 per kg,
        // mixing price 20
        return a.getUnitPrice()*0.4 + c.getUnitPrice()*0.6 +
        20 + this.q3 * 300;
    }
    @Override
    public void serve() {
        System.out.println("Packaging
        HoneyNut of " + this.getQuantity() + " kg.");
        this.a.serve();
        this.c.serve();
        System.out.println("Stir nuts.");
        this.addHoney(this.q3);
        // Mixing nuts
        System.out.println("Stir nuts.");
        System.out.println("Add spice.");
        System.out.println("Stir nuts again.");
    }
}

public class Smells {
    public static void main(String [] args)
    {
        MixedDriedNuts mixedDriedNuts = new
        MixedDriedNuts(1.0);
        mixedDriedNuts.serve();
        System.out.println("Price " +
        mixedDriedNuts.getPrice());
        HoneyNut honeyNut = new HoneyNut(2.0, 0.5);
        honeyNut.serve();
        System.out.println("Price " + honeyNut.getPrice());
    }
}

```

=====