



INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

End-Autumn Semester Examination 2023-24

Date of Examination: _____ Session: (FN/AN) _____ Duration: 3 Hours Full Marks: 100

Subject No. : CS60059 Subject : Object-Oriented Software Department: CSE

Specific charts, graph paper, log book etc., required: NIL

Important Instructions:

- Answer all questions
- All answers should be brief and concise. Lengthy, ambiguous, and irrelevant answers will be penalized.
- No clarifications to any question shall be provided. In case you have any doubts, you can make suitable assumptions, but please write down your assumptions clearly.

1. Develop a sequence diagram to depict the interactions among various objects that take place when the doPayroll method is invoked for the following Java code. [4]

```
public class Payroll {
    private PayrollDB itsPayrollDB;
    private PayrollDisposition itsDisposition;
    public void doPayroll() {
        List employeeList = itsPayrollDB.getEmployeeList();
        for (Iterator iterator = employeeList.iterator(); iterator.hasNext();) {
            String id = (String) iterator.next();
            Employee e = itsPayrollDB.getEmployee(id);
            if (e.isPayDay()) {
                double pay = e.calculatePay();
                double deductions = e.calculateDeductions();
                itsDisposition.sendPayment(pay - deductions);
            }
        }
    }
}
```

2. Assume that you have to design and implement a file system. The main items that can be stored in the file system are files and directories. Each directory can contain zero or more files and zero or more directories. You should treat directories and files in a uniform way. For example, both will have a name and a double click on a file icon would open the file and a double click on a directory will show the contents of the directory.
- a) Which design pattern should be used to effectively implement the required file system? [1 mark]
 - b) Provide UML class design for your solution. [5 marks]
3. Consider that a financial application which you have developed uses a legacy account class named **CBal** which supports a method named **computeBalance()** to compute the cash balance. It has come to your notice that a new and more efficient account class named **NewCBal** which supports an equivalent method **cashAtHand()** has become available. However, you do not want to change the code of your financial application. Instead, you plan to design an object adapter for using the new account class. Give the class diagram for your object adapter solution. Under which situation would it be a better idea to use a class adapter? [5+1]
4. Consider the following Java code. The **getInternetAccess()** method of the **InternetAccess** class is invoked by the clients needing Internet access.
- a) Draw the class diagram for the given code. [2]
 - b) In a sequence diagram represent the method interactions that take would place upon invocation of the **getInternetAccess()** method. [3]
 - c) Name the GoF design pattern that has been implemented in the code. [1]


```
public interface OfficeInternetAccess {
    public void grantInternetAccess();
}

public class RealInternetAccess implements
OfficeInternetAccess {
    public void grantInternetAccess() {
        System.out.println("Internet Access granted");
    }
}
```

```
public class InternetAccess implements OfficeInternetAccess {
    private RealInternetAccess realAccess;
    public void getInternetAccess() {
        if (getJobLevel(employeeId) > 4) {
            realAccess = new RealInternetAccess(employeeName);
            realAccess.grantInternetAccess();
        }
        else
        {
            System.out.println("Internet access NOT granted.");
            System.out.println("Your job level is below 5");
        }
    }
}
```

5. Draw the class diagram for the following Java program. Other than the iterator pattern, which GoF design pattern has been used in the given code. Identify the classes that participate in your identified pattern. [5+1]

```
class AgeComparator implements Comparator{
    public int Compare(Object o1, Object o2){
        Student s1=(Student)o1;
        Student s2=(Student)o2;
        if(s1.age==s2.age)
            return 0;
        else if(s1.age>s2.age)
            return 1;
        else
            return -1;
    }
}
```

```
class Student{
    int rollno;
    String name;
    int age;
    Student(int rollno, String name, int age){
        this.rollno=rollno;
        this.name=name;
        this.age=age;
    }
}
```

```
class Simple{
    public static void main(String args[]){
        ArrayList al=new ArrayList();
        al.add(new Student(101, "Vijay", 23));
        al.add(new Student(106, "Ajay", 27));
        al.add(new Student(105, "Jai", 21));
        System.out.println("Sorting by Name...");
        Collections.sort(al, new NameComparator());
        Iterator itr=al.iterator();
        while(itr.hasNext()){
            Student st=(Student)itr.next();
            System.out.println(st.rollno+" "+st.name+" "+st.age);
        }
        System.out.println("sorting by age...");
        Collections.sort(al, new AgeComparator());
        Iterator itr2=al.iterator();
        while(itr2.hasNext()){
            Student st=(Student)itr2.next();
            System.out.println(st.rollno+" "+st.name+" "+st.age);
        }
    }
}
```

```
class NameComparator implements Comparator{
    public int Compare(Object o1, Object o2){
        Student s1=(Student)o1;
        Student s2=(Student)o2;
        return s1.name.compareTo(s2.name);
    }
}
```

6. Assume that an educational ERP system has been written in Java. The programmer has used a static array `students` to stored the `Student` objects. It is required to implement an iterator for the student array. For this, give your design in the form of a UML class diagram. [6]

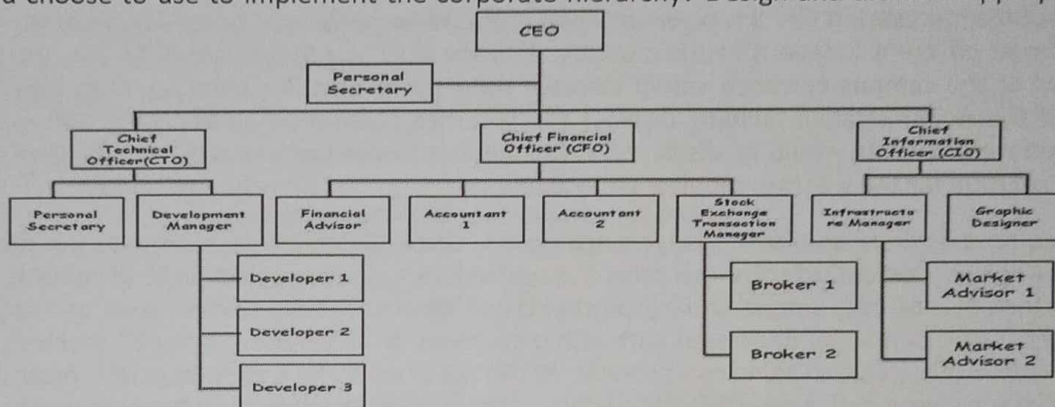
```
public class Studs{
    private Student [] students=new Student[5000];
    ...}
}
```

7. An intelligent war game software needs to be designed. Using this gaming software, the army officers of a country can practice war scenarios. Each side of a game consists of a large number of configured soldiers and captains, but a single general. Any war move by a soldier needs to be intimated to the fellow soldiers and the assigned captain. A soldier may communicate important observations regarding the war first to the assigned captain, who may in turn communicate it to the general if he deems it important. The general may issue commands, which are broadcast to all his captains and soldiers. As the game proceeds, soldiers may die or may get captured by the enemy side and these events need to be intimated to all players from both sides. Identify the applicable design pattern and suggest a class structure that would form the core of the war game software. [6]

8. Oak Tree company wants to develop a software to be called **Arena** that is intended to help improve social interactions among its employees. Employees of the company can play online games using **Arena**. Initially **Tic Tac Toe (TTT in short)**, **Chess**, **Snake and Ladder (SL in short)** and **Checker** would be supported. Later more games would be added. Any two employees may instantiate any one of the supported games at any time. A game instantiation consists of displaying the board in the computer terminals of the two players, instantiating the game control software, as well as instantiating the corresponding statistics gathering software. An employee should be

able to play any of the supported games and for any number of times. For each of the supported games, the company intends to award cash prizes to the employee winning against maximum number of opponents. Design the class diagram for **Arena** game software. (Hint: Use Abstract Factory pattern) [6]

9. Suppose a company has asked you to implement a software that would help it to automate its employee management activities such as leave approval, and handling various types of reimbursements. The reimbursements are considered as privileges and are employee-specific, therefore the software should support enabling and disabling various privileges granted to an employee. All employee management activities should be based on a hierarchical reporting structure as shown in the following diagram. Besides the top level managers: CEO, CTO, CFO, and CIO, the managerial cadre also includes the technical managers: Development manager, Stock Exchange manager, and Infrastructure manager. A technical manager is an employee who leads engineers, technicians, support personnel, all of whom are non-managerial employees, and also some junior technical managers. A top-level manager usually leads some technical managers and some employees. Which design pattern would you choose to use to implement the corporate hierarchy? Design and draw an appropriate class diagram. [1+5]



10. An environmental monitoring application was developed few years back and is working satisfactorily. It uses more than a dozen modems of various make to connect to the data collection stations. All modem drivers conform to the **Modem** interface. The class design for this application is given in the following Fig. A. Now there is a requirement for a few more data collection station centres to be implemented that would collect same data (e.g. rainfall) as well as a few different types of data such as seismic activity. A new kind of modem (dedicated modem) is proposed to be used for these planned stations that would need no dialling to establish a connection. A new application program to process the new types of data from the new collection stations would be written. However, the existing application would also need the traditional data collected at the planned stations. In this situation, the ideal solution could have to segregate the Modem interfaces as shown in Fig. B (The new application is shown as Ded Users). However, this solution would require changing the code of the satisfactorily working application. This is unacceptable to the organization. Suggest how the dedicated modem can be supported without breaking the code of the working application or the modem driver. Name the applicable design pattern and give class diagram for your suggested solution. [1+5]

Fig. A

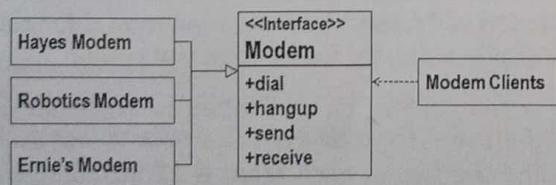
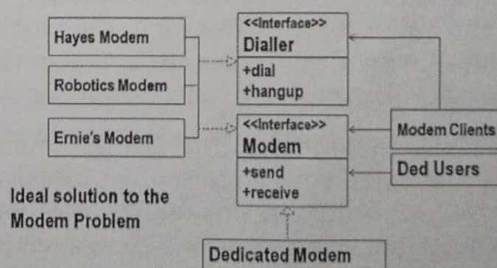


Fig. B



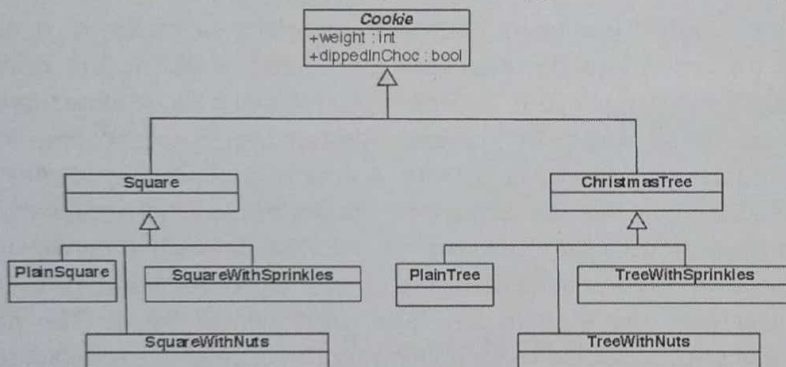
11. Suppose you need to write an application to simulate an electronic system. An electronic system consists of several electronic components. Each electronic component is made of a few resistors, capacitors, **FLIP FLOPs**, **NAND**, and **AND** gates, and other electronic components. Each **NAND** gate consists of 2 **AND** gates. Each **FLIP FLOP** consists of several **NAND** and **AND** gates. Name the specific GoF pattern that is applicable. Draw a class diagram to represent an electronic system which would form the core of the design of the simulator. [1+5]

12. A multi-threaded application program that is being developed should have only one instance of the following **GameTable** class whose code is given below. It needs to be ensured that only a single **GameTable** object is ever created and accessed by the other objects in the application by calling **GameTable.getInstance()**. Write the Java code for the singleton version, and clearly indicate what code you have added or modified. [6]

```
public class GameTable {
    private int limit;
    public GameTable() {
        this.limit = 0;
    }
    public void setLimit(final int limit) {
        this.limit = limit;
    }
    //Other game table operations. . .
}
```

13. Consider that a set of sensors are to be placed in the IIT(Kgp) weather monitoring station (in front of CSE Department) to sample various weather parameters such as temperature, pressure, humidity, precipitation, and wind speed. A weather monitoring application would be developed in Java to run on a desktop placed at the weather monitoring station. Whenever a significant variation over the last reported reading occurs, the corresponding sensor invokes the **report** method of the **WeatherMonitor** class. The weather data are to be displayed real-time at different locations in the campus. Also the display formats are different, for example a large display terminal at the campus entrance would display the current weather parameters in Graph. A display terminal in front of the Administrative building displays the weather parameters in text. It is anticipated that in future the collected weather data would be displayed at many more places and in novel formats. Give a class diagram for your design solution for the weather monitor application. [6]

14. It is required to develop a software package called **CookieCutterPro** to automate the production of a variety of baked Cookies. **CookieCutterPro** can drive a machine based on user selection to produce cookies of various shapes which include not only simple squares and circles, but also special shapes, such as Christmas trees and Easter bunnies. Cookies may be decorated with either sprinkles or nuts. Some types of cookies are dipped in chocolate. Every cookie is in addition to its composition, is also specified by its weight in grams. After this requirement for the software was expressed, a programmer suggested the following fragment of the core design:



- a) Write down the specific problems (if any) in the programmer's solution. [1]
 b) Give an alternative design to overcome the identified shortcomings in the form of a UML class diagram. Explicitly name any design patterns or principles you have chosen to apply. [4+1]

15. A security system has been designed using a light with switch and a motion sensor. The motion sensor can be either armed or unarmed. When the switch is in the **OFF** position, the light is **OFF** and the motion sensor is unarmed. When the switch is made **ON** the light stays **OFF**, but the motion sensor is armed. When the motion sensor is armed and it detects any movement, then the light comes **ON**. If no movement is detected for 5 seconds, the light goes **OFF**, otherwise light stays **ON** and alarm is sounded. A) Draw state machine model for the security system. B) Design class diagram using state pattern for the controller software for the security system. [4+4]
16. Consider the following excerpt from the SRS document for a software to be developed to automate the book-keeping activities associated with arranging the logistics for the conduction of the 2024 FIFA World Cup Football tournament. In this tournament, teams represent countries and the size of each team is 22 including the players and the officials accompanying the team. The games are to be conducted in a number of selected cities of the designated country. Each team should play at most two games in any one city. The set of referees would be drawn such that each participating country should have at least one referee. Two Referees are to be assigned to each game. Hotel reservations are to be made for the referees and the team members for the scheduled tournament days in one or more hotels in the city where the teams are scheduled to play. Identify the domain entity classes and the identifiable relations among them and represent those in a UML class diagram. No need to represent boundary and controller classes. [10]

--- The End ---