

Roll No. 20L-1133Name HuzaiFa Bin Faheem**Quiz 6**
Weight: 2%

Q1. Explain how a design class diagram (DCD) is derived from an analysis class diagram (ACD).

1

An ACD is used to derive DCD as all the class attributes and class functions made in ACD are given data types and assigned one of the three class scopes. Hence a DCD is built upon an existing ACD and derived from it.

Q2. Explain how a software product's DCD and its design sequence diagrams (DSDs) are connected to each other.

0.5

The DCD represents the classes, and shows their relations. The DCD also defines data types of the attribute of each class. DSDs defines the operations made by and upon the objects of those classes in DCD, and shows all possible courses of actions that can be made by those classes.

Q3. Assume that you are given a complete set of (tabular template-based) use cases (UCs) of a software product. Also assume that you are **not** allowed to use the "alt" interaction frame in DSDs. In such a situation, how would you determine the total number of DSDs to create for that software product?

0.85

All ~~possible~~ UCs and their ~~possible~~ ~~course~~ alternate course of actions should be represented in the DSD. Total number of DSDs would be (number of UCs) + ~~(~~alt~~ number of)~~ (sum of alternate course of actions of all UCs)

Fall 2022

Roll No. 20L-1133

1

Name Huzaira Bin Fabeen**Quiz 8**

Weight: 1%

```

class book
{
    char bno[6];
    char bname[50];
    char aname[20];
public:
    void create_book()
    {
        cout<<"\nNEW BOOK ENTRY...\n";
        cout<<"\nEnter The book no.";
        cin>>bno;
        cout<<"\nEnter The Name of The Book ";
        gets(bname);
        cout<<"\nEnter The Author's Name ";
        gets(aname);
        cout<<"\n\nBook Created..";
    }

    void show_book()
    {
        cout<<"\nBook no. : "<<bno;
        cout<<"\nBook Name : ";
        puts(bname);
        cout<<"Author Name : ";
        puts(aname);
    }
}

```

```

void modify_book()
{
    cout<<"\nBook no. : "<<bno;
    cout<<"\nModify Book Name : ";
    gets(bname);
    cout<<"\nModify Author's Name of Book : ";
    gets(aname);
}

char* retbno()
{
    return bno;
}

void report()
{
    cout<<bno<<setw(30)<<bname<<setw(30)<<
    aname<<endl;
}

}; //class ends here

```

Q. Determine the value of CK metric LCOM for the book class* given above. Complete working (i.e. all steps) must be shown to get credit. You may use this paper's back side if necessary.
 *(code taken from <http://www.cppforschool.com/project/library-management-system.html>)

$$I = \{ bno, bname, aname \}$$

$$M = \{ create_book, show_book, modify_book, retbno, report \}$$

$$I_1 = \{ bno, bname, aname \}$$

$$I_2 = \{ bno, bname, aname \}$$

$$I_3 = \{ bno, bname, aname \}$$

$$I_4 = \{ bno \}$$

$$I_5 = \{ bno, bname, aname \}$$

$$P = \{ \emptyset \}$$

$$Q = \{ (I_1, I_2), (I_1, I_3), (I_1, I_4), (I_1, I_5), \\ (I_2, I_3), (I_2, I_4), (I_2, I_5), \\ (I_3, I_4), (I_3, I_5), \\ (I_4, I_5) \}$$

$$LCOM = |P| - |Q|, \text{ when } Q > P, LCOM = 0$$

Hence $LCOM = 0$

$$|P| = 0$$

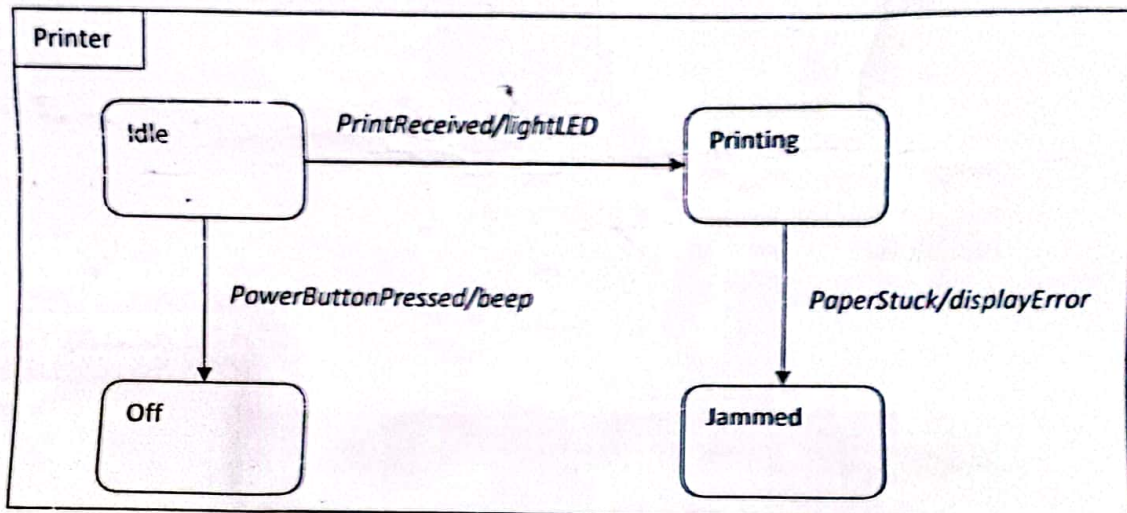
$$|Q| = 10$$

Fall 2022

Roll No. 202-1133Name Huzai'fa Bin Fateem**Quiz 5**

Weight: 1%

The following depicts a defective state diagram for objects of the Printer class. List 5 different types of non-semantic defects in this diagram along with their respective rectification in the table given below.



S#	Defect : Rectification
1	Defect: Class name box has incorrect number of edges Rectification: It should be Printer Printer ✓
2	Defect: Arrow heads are filled Rectification: Arrow heads should be "→". ✓
3	Defect: Activities are in italics Rectification: Activities should not be in italics, should be normal font style. ✓
4	Defect: No start and end "•" and "○" even though it's not a state diagram. Rectification: Make start or end or make diagram a known state diagram. ✓
5	Defect: Rectification:

Roll No. 20L-1133Name Huzair Bin Fahim**Quiz 7**
Weight: 1%

Q1. Why is over-modularization bad?

The greater the number of modules, the greater the cost for integration. More modules \Rightarrow greater cost for integration.

Q2. Why is it important to design classes that are highly cohesive?

So that if the code needs to be maintained or extended, a person with little to no knowledge of the entire program code can update a functionality without wasting time and resources to first understand the code for the entire program.

Q3. Suppose function f1 simply computes the strike rate of a cricket batter using the following formula and then returns the result. What type of cohesion is exhibited by f1?

$$\text{strike_rate} = (\text{runs_scored} / \text{balls_faced}) * 100$$

Functional cohesion.

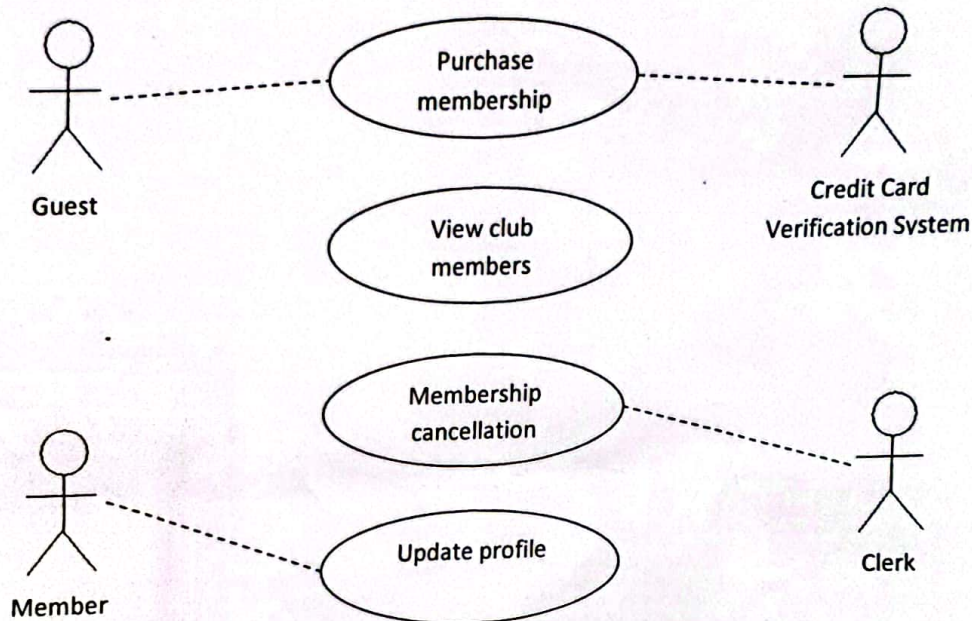
Q4. Why should classes be open for extension but closed to modification (Reference: O in SOLID)? Explain.

~~So that if~~ If a class is open to ~~modification~~ modification and is used in other parts of a program, whenever the class is modified, its functionality will be needed to be checked ~~everytime its modified~~ everywhere ~~the~~ ^{object} class is used. Hence ~~it~~ class should be closed to modification and only open to extension in case additional functionality is needed to be added.

Roll No. 20L-1133Name Huzefa Bin Faeem**Quiz 2**

Weight: 1%

The following depicts a defective use case diagram of a club membership management system. Identify 5 different types of defects in this use case diagram.

Club Membership Management System

S#	Defect
1	Participation Line is dotted instead of filled. ✓
2	The system is not in a rectangular box. ✓
3	View club members use case has no participating actors. Name of human actor is not specific ✓
4	Credit card verification system is non-human actor but has figure of human actor. ✓
5	Use cases ^{names} are not starting with down case letters. ✓

Fall 2022

Roll No. 20L-1133Name Huzafa Bin Fahem**Quiz-1**

Weight: 1%

Q1. What is the relationship between Computer Science and Software Engineering?

0.25 Computer Science is the study and theory involved behind different data structures and algorithms, whereas software engineering uses those data structures and algorithms to develop solutions to various problems using software.

Q2. How is object-oriented analysis different from object-oriented design with respect to scope?

0.20 Object-oriented analysis is the analysis of the problem for which software is to be made and involves the formation of domain, state and application models. Object-oriented design uses those models and implements the code with classes, inheritance and polymorphism.

Q3. Why is programming analogous to "donkey work"?

0.25 Before the programming phase begins, it is already decided in the prior phases what data structures, classes and algorithms are to be used.

Q4. How does a model help us in dealing with complexity?

0.20 A model is an abstraction of the problem which is free of ~~now~~ different implementations and only provides an overview. Helping us deal with complexity.

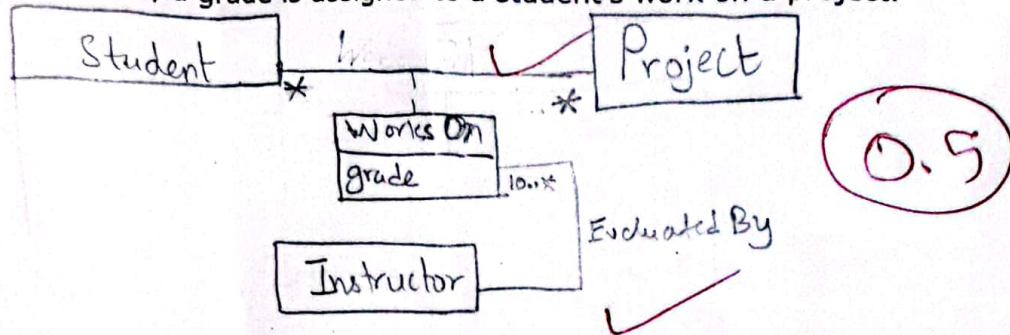
Fall 2022

Roll No. 20L-1133Name Huzafa Ben Fahim**Quiz 4**

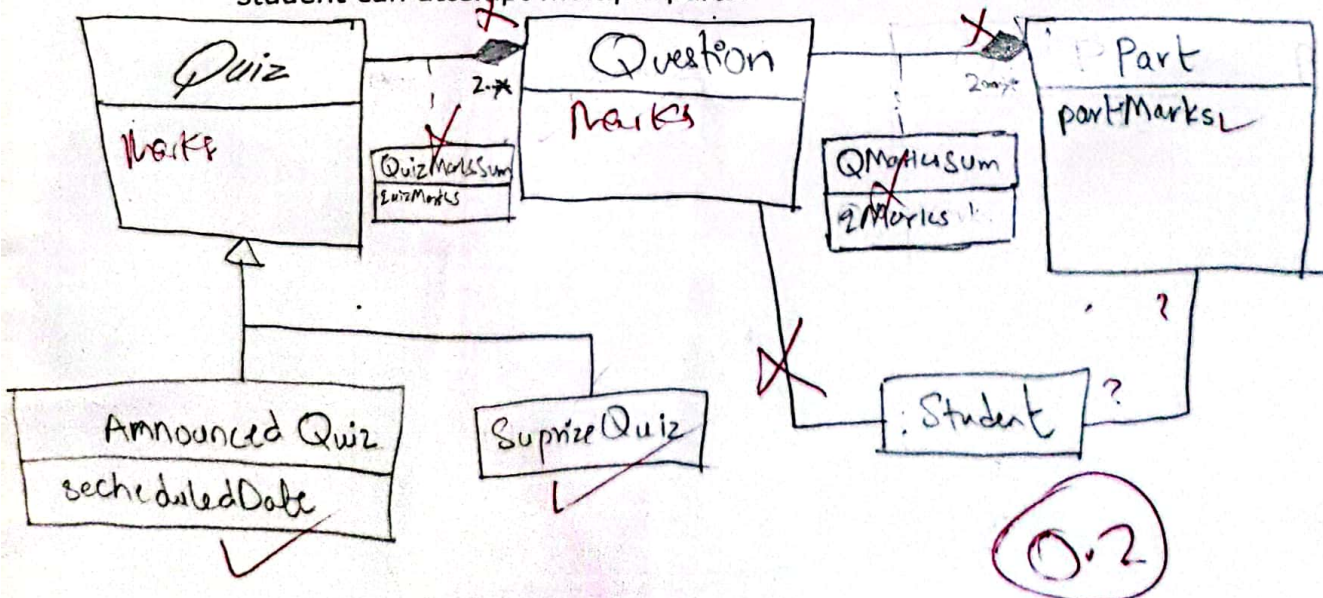
Weight: 1%

Without making any assumptions, model **just** the following information using a UML 2 analysis class diagram.

- a. A student can work on multiple projects. Each project can have multiple students working on it. A student's work on a project is evaluated by an instructor. Every instructor evaluates at least 10 works. After every evaluation, a grade is assigned to a student's work on a project.



- b. Quizzes are of exactly two types i.e. surprise and announced. Announced quizzes have a scheduled date. Each quiz contains at least two questions and each question contains at least two parts. Each part, question, and quiz has marks. Marks for a quiz are the sum of the marks of all questions in that quiz. Marks for a question are the sum of the marks of all the parts in that question. Each part can be attempted by multiple students. Similarly, each student can attempt multiple parts.



Fall 2022