Mid-2 Exam OOAD (Fall 12) Max Time 1:30
FAST-NU Lahore Max Marks: 40

Consider a web-based file storage and sharing system that allows its users to create folders, upload/download files, and share these with other users. A folder can contain any number of files and/or folders. Every user has a login and password. The user can share his/her files/folders with other users by changing file/folder permissions. Files/folders can be shared to all users or a selected subset of users. The users other than the owner cannot remove or rename files/folders however.

The system provides all the necessary functionality such as sign-up, or deleting or renaming a folder etc.

1. Give a use case diagram for the system. (10)
2. Give use case description for "Share File/Folder". Also provide descriptions of included use cases or extensions if any. (10)
3. Give a class diagram showing required attributes and functions. (20)

National University of Computer and Emerging Sciences

Course:	Software Design & Architecture	Course Code:	SE2002
Program:	BS (SE) 4A	Semester:	Spring 2022
Duration:	60 Minutes (1 Hour)	Total Marks:	20 (16)
Paper Date:	06-May-22	Weight	15%
Section:	All	Page(s):	3
Exam:	Sessional II		

Instruction/Notes:

Attempt all questions on the question paper. Neither use nor submit any extra sheet.

Name: _____ Roll Numb_____ Section ___

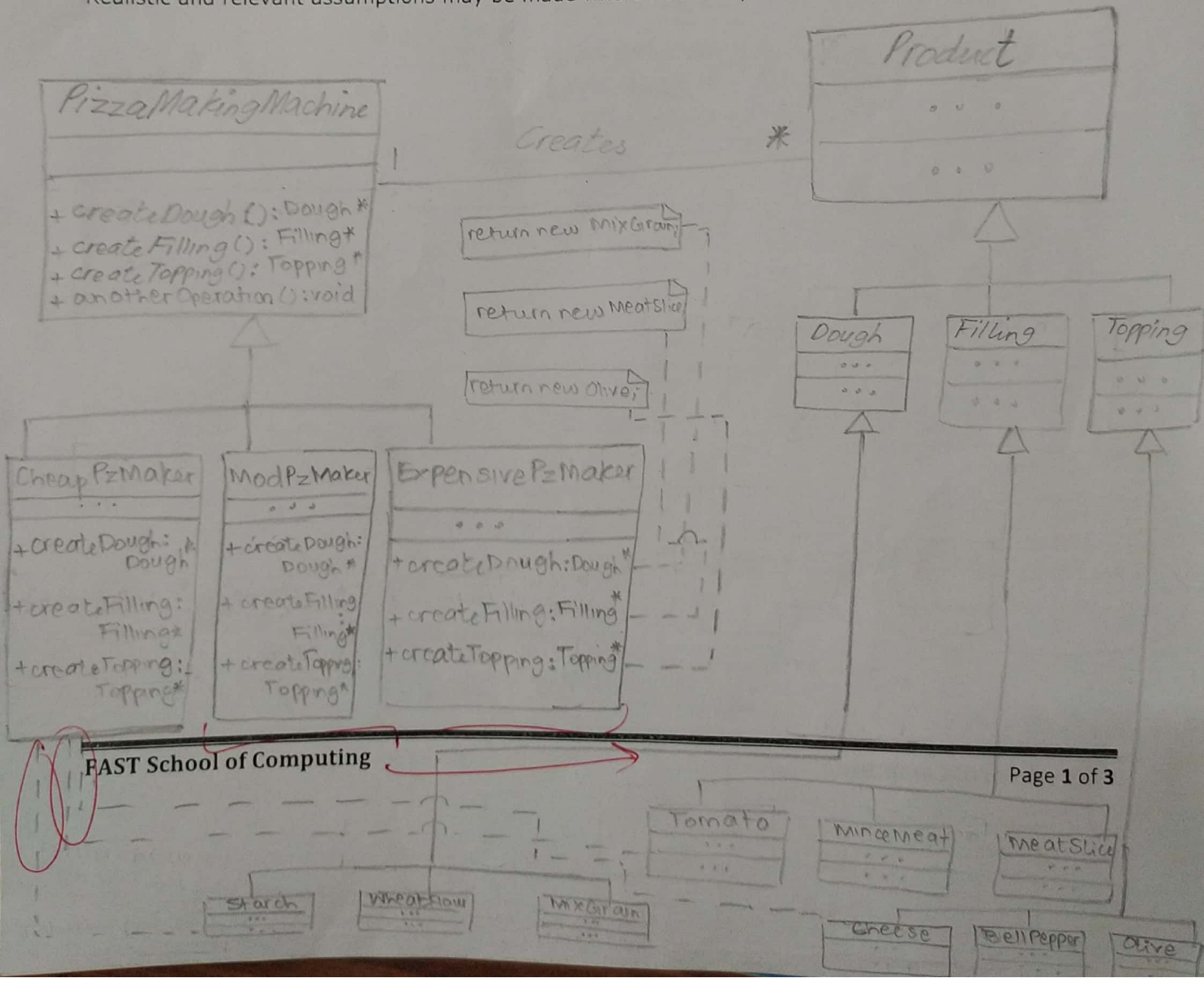
associations?

Question 1 (Max. Marks = 10)

(9)

Pizza making machines are of three types i.e. cheap pizza maker, moderate pizza maker, and expensive pizza maker. The cheap pizza maker makes a starch dough, a tomato filling, and a cheese topping. The moderate pizza maker makes a wheat flour dough, a minced meat filling, and a bell pepper topping. The expensive pizza maker makes a mixed grain dough, a meat slice filling, and an olive topping.

Map the information given above to a **UML 2 design class diagram** that uses the **Factory Method** design pattern. Your diagram should be annotated with relevant comments containing C++ code. Realistic and relevant assumptions may be made where necessary.

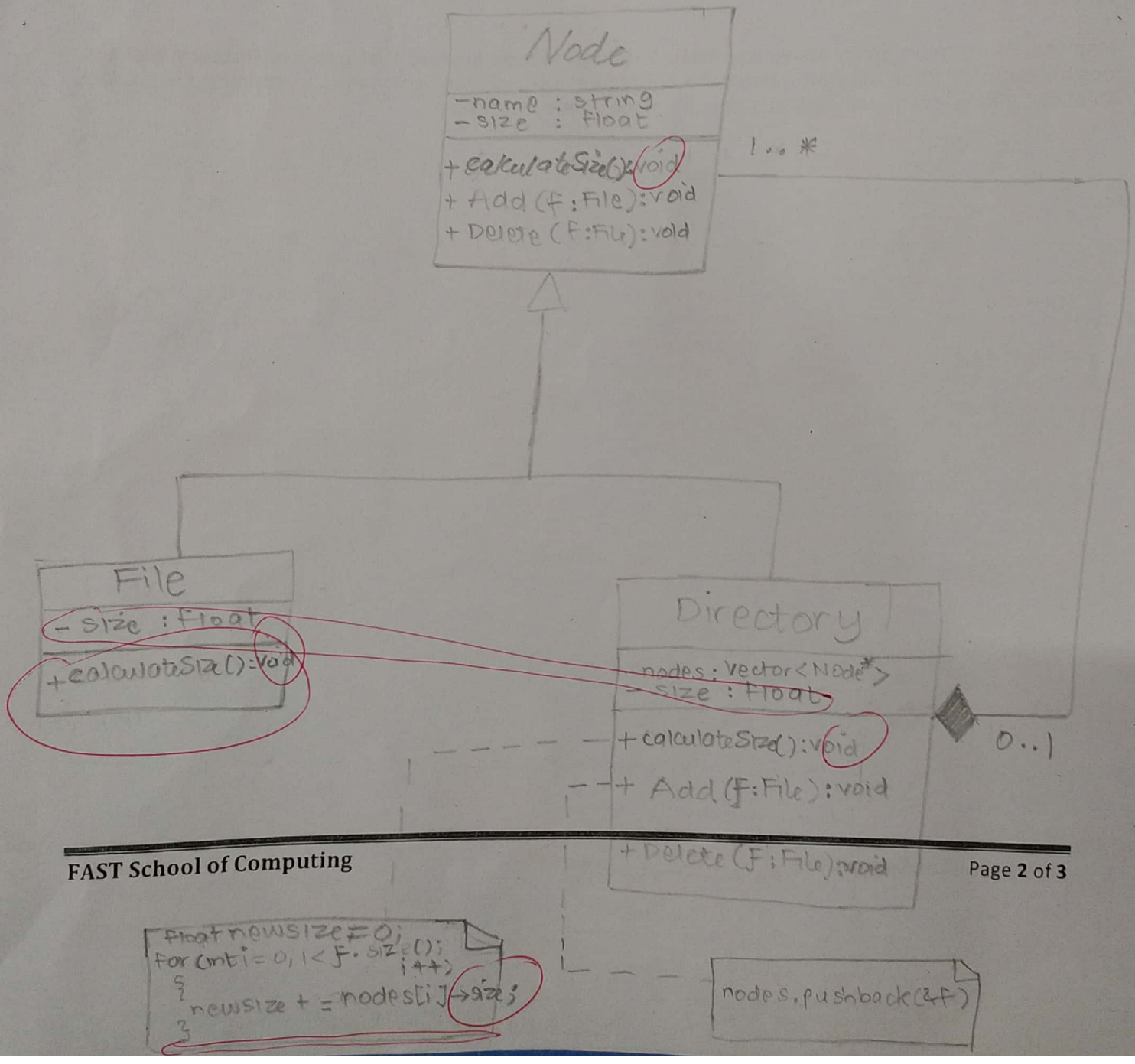


Question 2 (Max. Marks = 10)

Consider a simple file system. A node in the file system is either a file or a directory. A directory may contain one or more files. It may also contain one or more other directories. Every node has a name and size. The size of a directory is equal to the sum of the sizes of all the files and directories contained in it.

a. Which design pattern is applicable for the file system described above? [Note: An invalid answer to this part will result in a zero in the entire question.] composite

b. Draw a **UML 2 design class diagram** showing the design of the file system described above. This design **must** adhere to the structure of the design pattern chosen in part a above. Also, your diagram should be annotated with relevant comments containing C++ code. Realistic and relevant assumptions may be made where necessary.



CLO 3: Implement object-oriented principles for software analysis and design

Q1 ..... [10 x 2 = 20]

a. Consider the following design. Identify which SOLID principle is violated and why. Propose a design to fix the violation.
Note: No credit will be given in case of failure to identify the most appropriate SOLID principle (being violated).

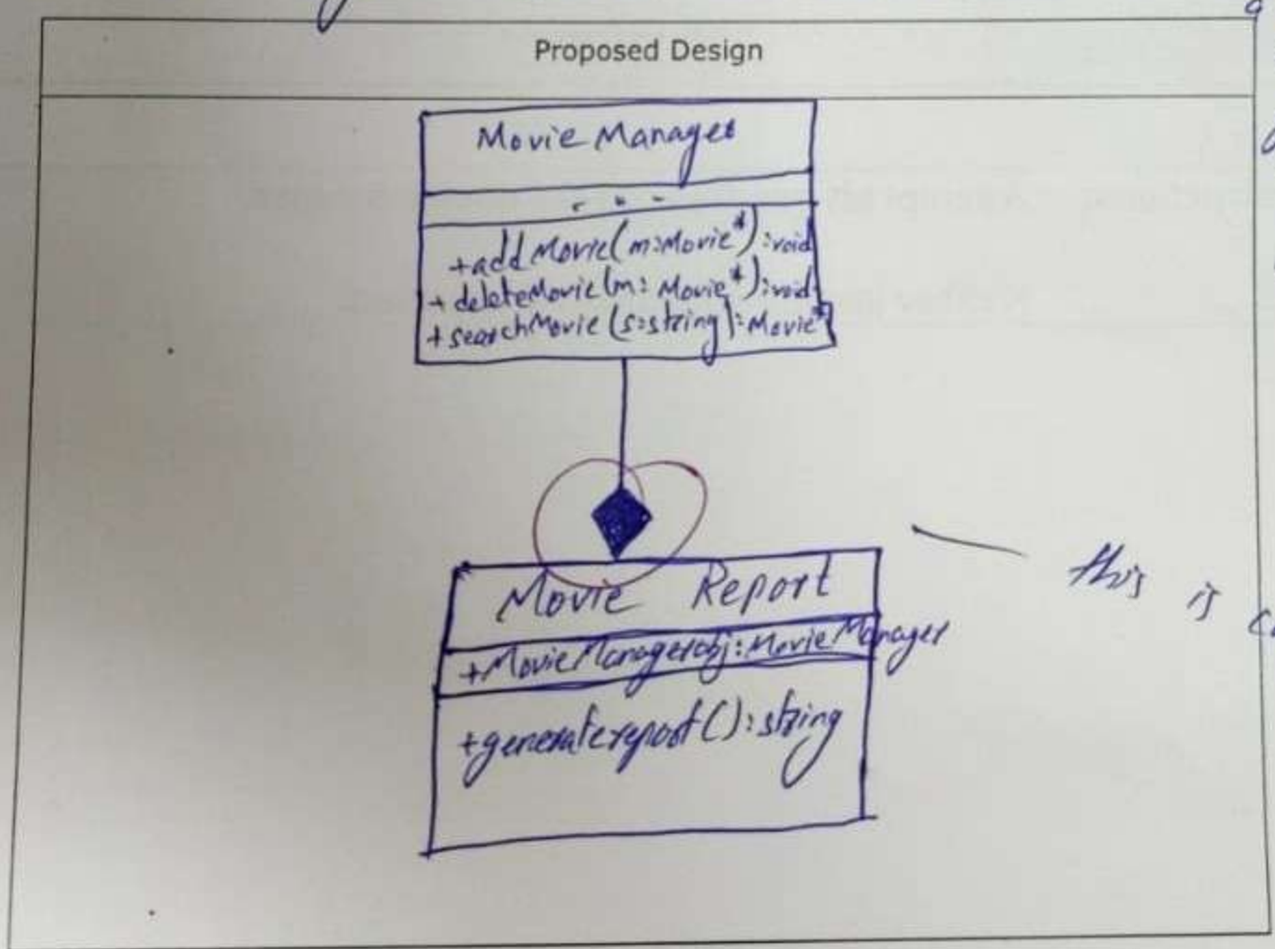
MovieManager
+addMovie(m: Movie*): void +deleteMovie(m: Movie*): void +searchMovie(s: string): Movie* +generateReport(): void

8

SOLID principle being violated: Separation of functions/concerns. (1st poircipl of Solid Principles).

Reason for violation (less than 40 words):
Each class should have a very specific function and not only adds/deletes/ searches the movies, it also generates a report. The report generation should be in a different class.

Proposed Design



b. Consider the following code. Identify which SOLID principle is violated and why. Modify the code to fix the violation.
Note: No credit will be given in case of failure to identify the most appropriate SOLID principle (being violated).

```
public class Account {
  protected double balance;
  public void deposit(double
amount) {
    if (amount <= 0) {
        throw new
        IllegalArgumentException
        ("amount <=0");
    }
    balance += amount;
  }
}
```

```
public class SavingsAccount extends Account {
  @Override
  public void deposit(double amount) {
    super.deposit(amount);  // Call base class deposit
    // Apply minimum balance fee after successful deposit
    if (balance < 100) {
      // Charge a fee if balance falls below minimum
      balance -= 5;
    }
  }
}
```

3

SOLID principle being violated: Liskov Substitution Principle.

Reason for violation (less than 40 words):
Because the sub or child class needs to be replacable with the function of the parent/super class, but in this case the 2nd deposit function is called from the 1st deposit function

Modified Code

```
public class Account {
  protected double balance;
  public void deposit(
      double
      amount)
  {
    if (amount <=0)
    { throw new
      IllegalArgumentException
      ("amount <=0");
    }
    balance += amount;
  }
}
```

```
public class SavingAccount extend Account
{
  @Override
    public void deposit(double amount)
    {
      if (amount <=0) {
          throw new IllegalArgumentExcep
          ("amount <= 0");
      }
      balance += amount;
      if (balance <100) {
          balance -= 5;
      }
    }
}
```


0155

CLO 3: Implement object-oriented principles for software analysis and design

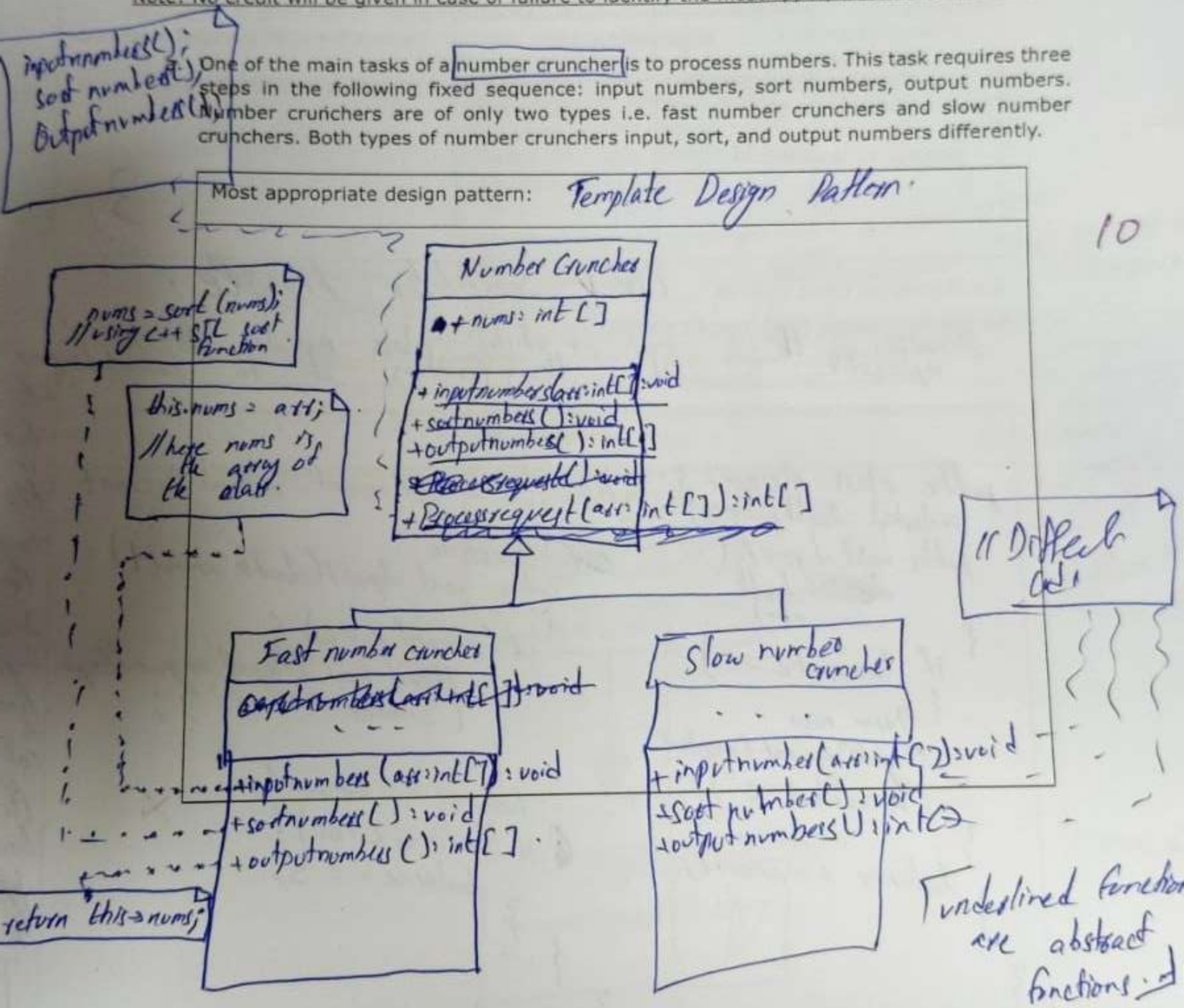
Q2 ..... **[10 x 2 = 20]**

Map the information given in each part below to a **UML 2 design class diagram** that uses the **most appropriate design pattern**. Annotate your diagram (drawn inside the box) with important comments containing error-free C++ code. Realistic and relevant assumptions may be made where necessary.
Note: No credit will be given in case of failure to identify the most appropriate design pattern.

a) One of the main tasks of a number cruncher is to process numbers. This task requires three steps in the following fixed sequence: input numbers, sort numbers, output numbers. Number crunchers are of only two types i.e. fast number crunchers and slow number crunchers. Both types of number crunchers input, sort, and output numbers differently.

Most appropriate design pattern: Template Design Pattern.

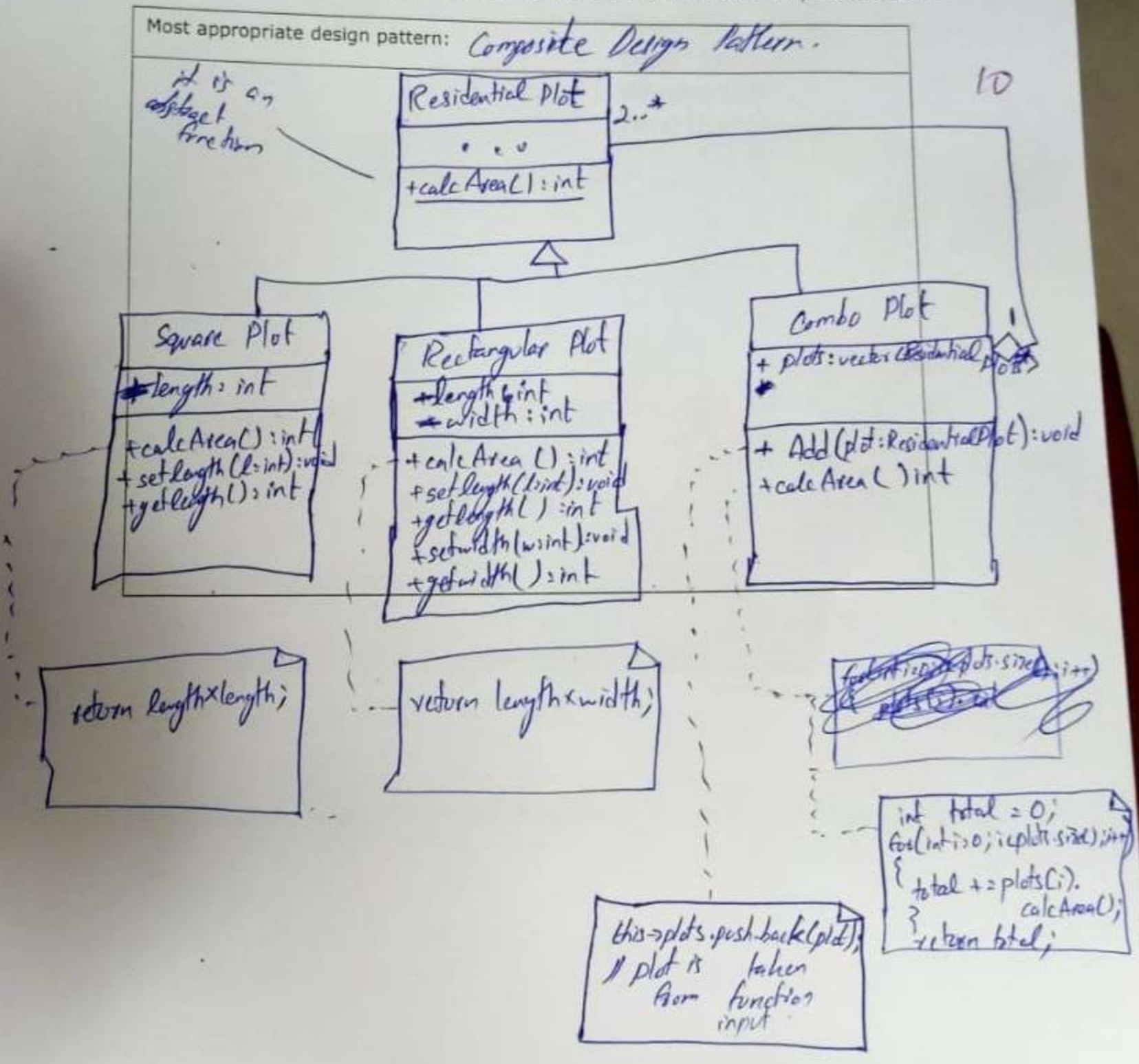
10



01

b. Residential plots are of only three types i.e. square plots, rectangular plots, and combo plots. A combo plot contains at least two residential plots of any type (including combo plots). Area can be calculated for all types of residential plots. The area of a square plot is the square of its length. The area of a rectangular plot is the product of its length and width. The area of a combo plot is the sum of the areas of the residential plots it contains.

Most appropriate design pattern: Composite Design Pattern.



National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**60 Minutes**	**Total Marks:**	**25**
Paper Date:	**16-Nov-18**	**Weight**	**15 %**
Section:	**All**	**Page(s):**	**6**
Exam:	**Midterm-II**	**Reg. No.**	

Instruction/Notes: Solve the exam on this paper. Do not submit answer sheets. You may use rough sheets but those shouldn't be attached.

Question 1 — 10 points

Consider a use case for a complaint / issue management system being used by an organization to ensure better customer support. In case of any problems, customers can register their complaints that can be tracked by the customers as well as the organization for efficient redressal.

UC: Register complaint
Purpose: To register complaint against a pending issue. Issues may be categorized depending upon their nature. For instance, a malfunctioning projector relates to support category while a slow internet connection relates to IT category. An issue may be simultaneously assigned multiple categories.
Actor: Customer
Main Flow:

1. Customer enters her identification details including mobile number and an optional email
2. Customer enters the issue detail as text
3. Customer selects appropriate categories from the available category list to mark the issue
4. Customer submits the form
5. System generates and shows a CAPTCHA code and ask the customer to enter the code, in order to prevent spurious complaints
6. Customer enters the code and submits the form
7. System registers the customer (if not already recorded in the system) and the complaint. A tracking number, timestamp, and status of the complaint along with given details and expected resolution time is shown.
8. System sends an email to the customer with the complaint details.
9. System routes the complaint to the concerned departments (based upon the category). A department may be responsible for multiple categories
10. Use case ends

Alternative Flow:

6A Customer may regenerate CAPTCHA code
7A If the CAPTCHA code is invalid, system regenerates the code, jumping to step-5 of main flow
8A If the customer didn't provide email address, system skips sending the email
9A If no category is assigned to the issue, system routes it to Support department by default.

Draw an activity diagram for the use case, illustrating possible concurrent activities

Course:	Object-oriented Analysis & Design	Course Code:	CS-309
Program:	BS (Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	25
Paper Date:	16-Nov-18	Weight	15 %
Section:	All	Page(s):	6
Exam:	Midterm-II	Reg. No.	

Course:	Object-oriented Analysis & Design	Course Code:	CS-309
Program:	BS (Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	25
Paper Date:	16-Nov-18	Weight	15 %
Section:	All	Page(s):	6
Exam:	Midterm-II	Reg. No.	

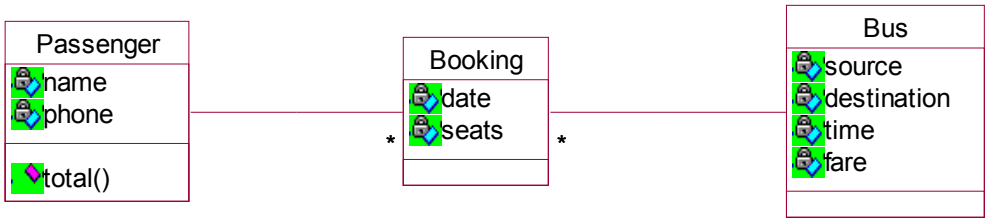
Question 2 **5 points**

Match each entry in the first column with a single entry in the second column. Try to give the best answer.

a) Suitable for concurrent tasks	1) Class diagram
b) Depicts the required object interactions	2) Sequence diagram
c) Useful for gathering requirements	3) Activity diagram
d) Shows static structure of the system	4) Use case diagram
e) Shows dynamic behavior of the objects	

Question 3 **10 points**

Consider the following class diagram:



Here the attribute "fare" holds the ticket-price, while the attribute "seats" denotes the number of seats reserved.

Now give a sequence diagram to compute the total payable amount for a passenger. The sequence diagram must include objects of above classes and message/function names with parameters and return value. You may add any auxiliary (helper) functions, such as getFare(). Give one-line description for each added function separately.

// sequence diagram

// one line description of functions

Course:	Object-oriented Analysis & Design	Course Code:	CS-309
Program:	BS (Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	25
Paper Date:	16-Nov-18	Weight	15 %
Section:	All	Page(s):	6
Exam:	Midterm-II	Reg. No.	

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**60 Minutes**	**Total Marks:**	**25**
Paper Date:	**16-Nov-18**	**Weight**	**15 %**
Section:	**All**	**Page(s):**	**6**
Exam:	**Midterm-II**	**Reg. No.**	

Instruction/Notes: Solve the exam on this paper. Do not submit answer sheets. You may use rough sheets but those shouldn't be attached.

Question 1 — 10 points

Consider a use case for a complaint / issue management system being used by an organization to ensure better customer support. In case of any problems, customers can register their complaints that can be tracked by the customers as well as the organization for efficient redressal.

UC: Register complaint
Purpose: To register complaint against a pending issue. Issues may be categorized depending upon their nature. For instance, a malfunctioning projector relates to support category while a slow internet connection relates to IT category. An issue may be simultaneously assigned multiple categories.
Actor: Customer
Main Flow:

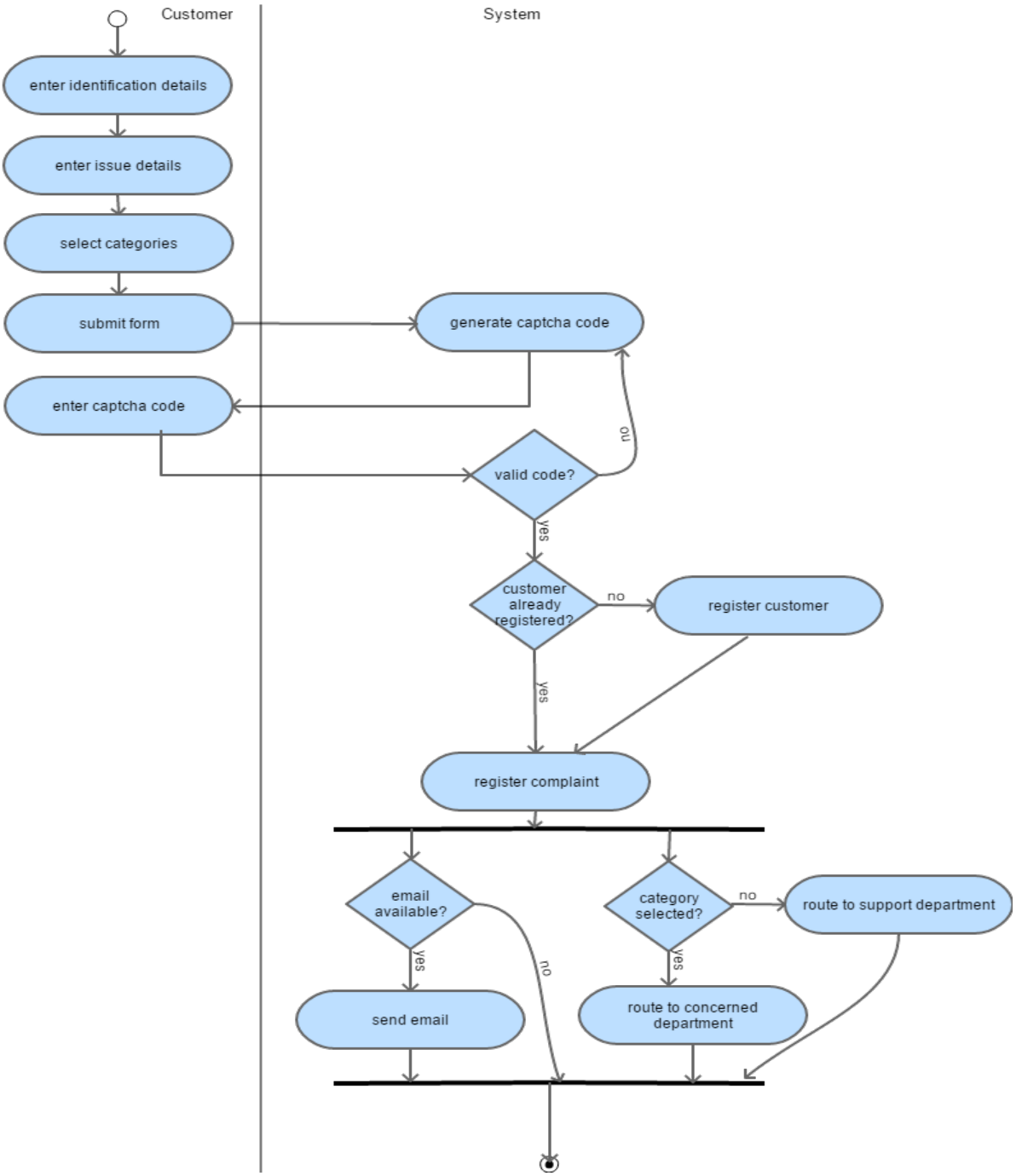
1. Customer enters her identification details including mobile number and an optional email
2. Customer enters the issue detail as text
3. Customer selects appropriate categories from the available category list to mark the issue
4. Customer submits the form
5. System generates and shows a CAPTCHA code and ask the customer to enter the code, in order to prevent spurious complaints
6. Customer enters the code and submits the form
7. System registers the customer (if not already recorded in the system) and the complaint. A tracking number, timestamp, and status of the complaint along with given details and expected resolution time is shown.
8. System sends an email to the customer with the complaint details.
9. System routes the complaint to the concerned departments (based upon the category). A department may be responsible for multiple categories
10. Use case ends

Alternative Flow:

6A Customer may regenerate CAPTCHA code
7A If the CAPTCHA code is invalid, system regenerates the code, jumping to step-5 of main flow
8A If the customer didn't provide email address, system skips sending the email
9A If no category is assigned to the issue, system routes it to Support department by default.

Draw an activity diagram for the use case, illustrating possible concurrent activities

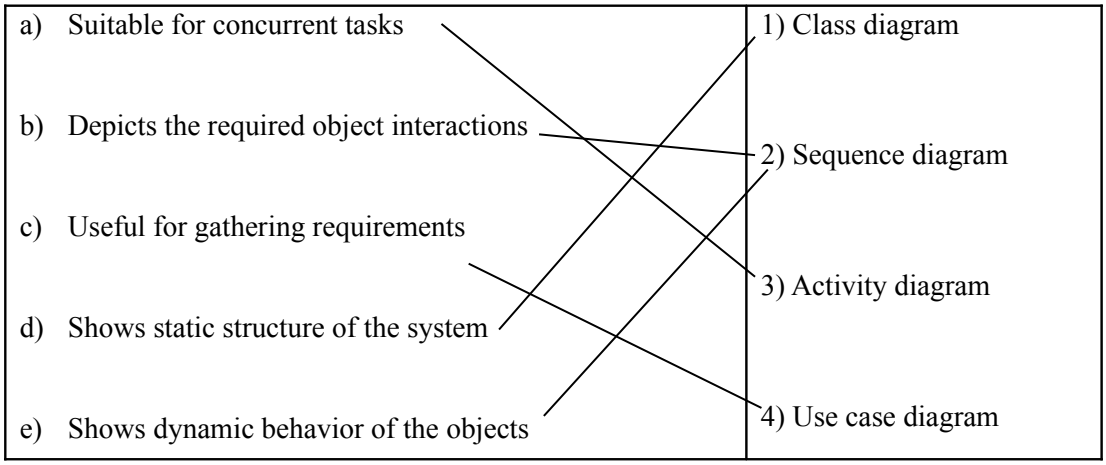
Course:	Object-oriented Analysis & Design	Course Code:	CS-309
Program:	BS (Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	25
Paper Date:	16-Nov-18	Weight	15 %
Section:	All	Page(s):	6
Exam:	Midterm-II	Reg. No.	



Course:	Object-oriented Analysis & Design	Course Code:	CS-309
Program:	BS (Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	25
Paper Date:	16-Nov-18	Weight	15 %
Section:	All	Page(s):	6
Exam:	Midterm-II	Reg. No.	

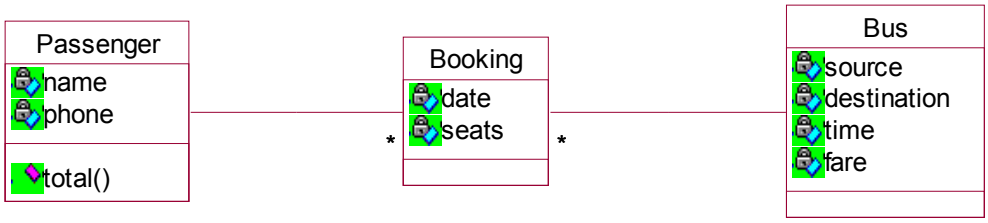
Question 2 **5 points**

Match each entry in the first column with a single entry in the second column. Try to give the best answer.



Question 3 **10 points**

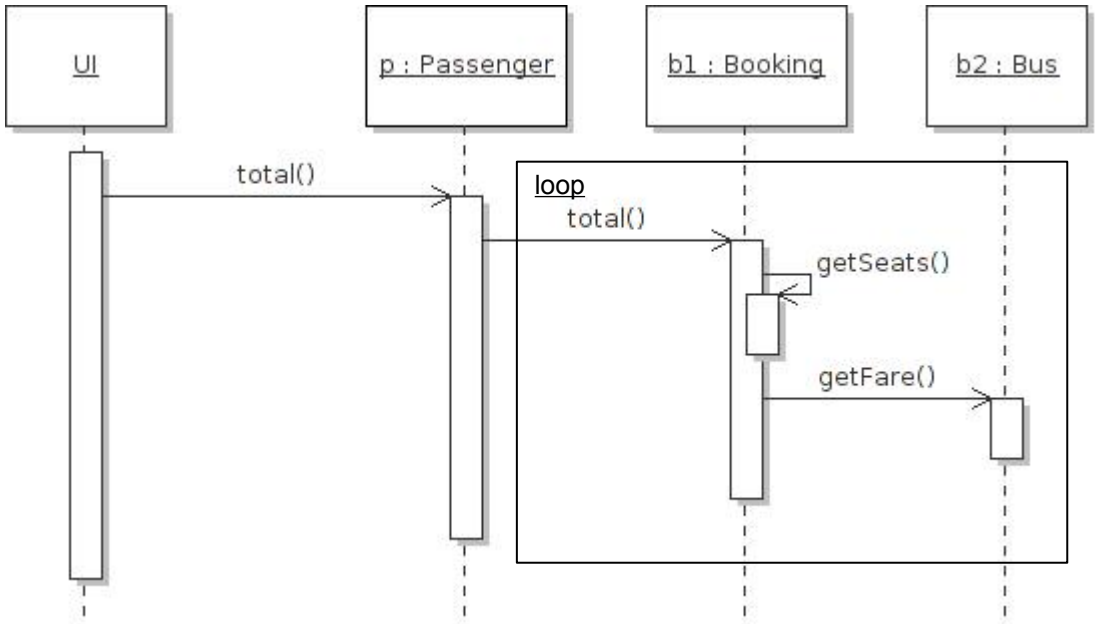
Consider the following class diagram:



Here the attribute "fare" holds the ticket-price, while the attribute "seats" denotes the number of seats reserved.

Now give a sequence diagram to compute the total payable amount for a passenger. The sequence diagram must include objects of above classes and message/function names with parameters and return value. You may add any auxiliary (helper) functions, such as getFare(). Give one-line description for each added function separately.

// sequence diagram



// one line description of functions

Bus::getFare: returns fare of bus

Booking::getSeats: returns number of seats reserved

Booking::total(): computes total for a single booking as product of seats and fare

Passenger::total(): computes total for the passenger, iterating over all bookings

National University of Computer and Emerging Sciences, Lahore Campus

Course:	Software Design & Analysis	Course Code:	CS3004
Program:	BS (CS)	Semester:	Fall 2023
Duration:	60 Minutes (1 Hour)	Total Marks:	30
Paper Date:	11-Nov-23	Weight	15%
Section:	All	Page(s):	4
Exam:	Sessional II		

Instruction/Notes:

Attempt all questions on the question paper. Neither use nor submit any extra sheet.

Name: SOLUTION Roll Number: _____ Section ____

Question 1 (Max. Marks = 15 = 5 + 10) [CLO 4]

A cricket-based mobile game - FASTCric - is designed in such a way that after a bowler delivers a ball to a batter, the batter's play function is called. This play function has two parameters i.e. ball type (leg spin, off spin, yorker, or bouncer) and ball speed (km/hour). This function decides which type of shot is played by the batter using the following rules:

- if leg spin or off spin is bowled, the batter plays the sweep shot
- if yorker is bowled, the batter plays the block shot
- if bouncer is bowled and ball speed is less than 80 km/hour and batter's energy level is at least 70%, then hook shot is played
- in all other cases, the leave-it-alone shot is played.

All types of shots keep track of the runs scored when they were played and the play function also returns the runs scored. A sweep shot always result in 4 runs while a hook shot always results in six runs. Both block shot and leave-it-alone shot result in 0 runs.

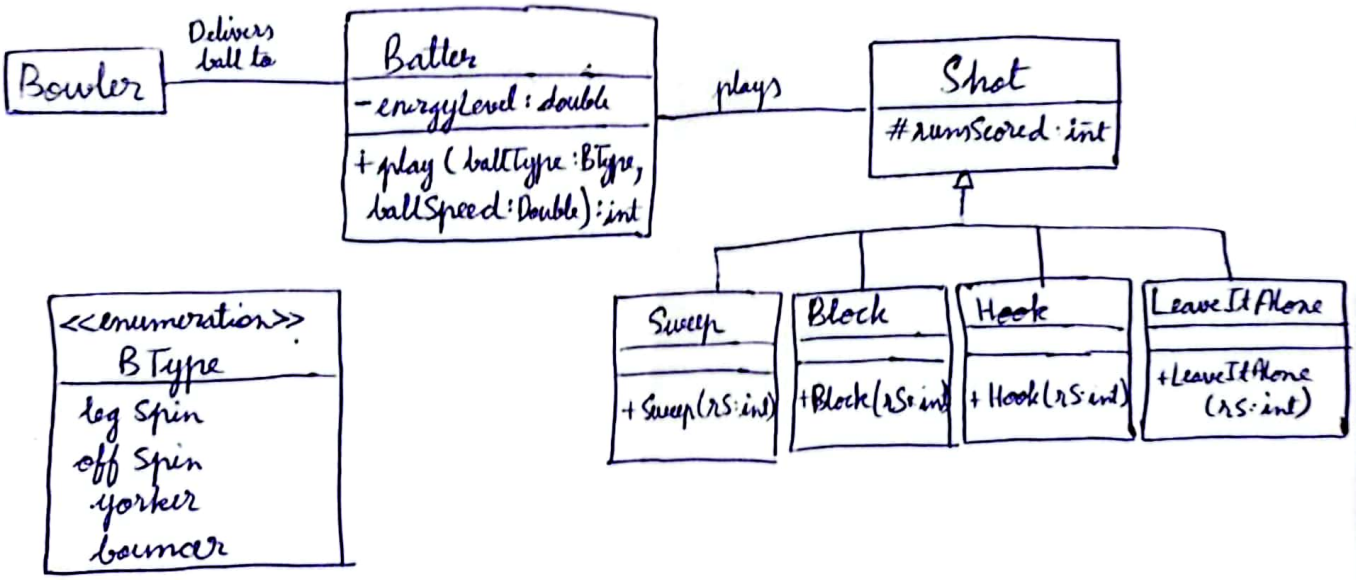
Model the information provided above using

a. a UML 2 **design class diagram** depicting a portion of the design of FASTCric
Important Instructions: This diagram should have exactly 7 classes (including Bowler, Batter, and Shot) and exactly 1 enumeration.

b. a UML 2 **design sequence diagram** depicting the "Play Ball" use case for the Batter actor.

Ensure consistency between these two diagrams.

[Use the space below on this page for answering Question 1a (UML 2 **design class diagram**) only.]

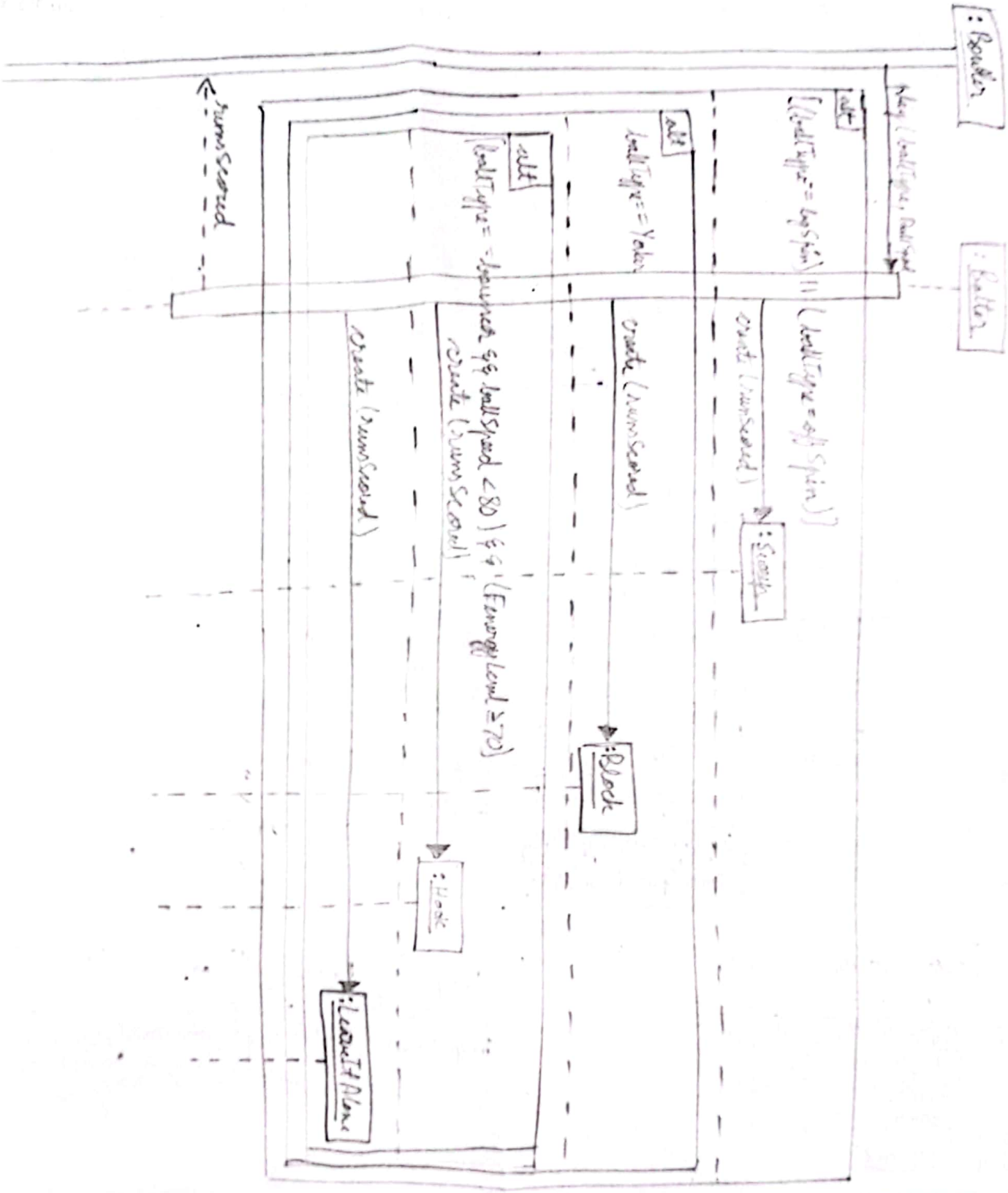


2+3 2.5+2.5

0011

Name: _____ Roll Number: _____ Section ____

[Use the space below on this page for answering Question 1b (UML 2 **design sequence diagram**) only.]



6 (Objects)
3 (Conditions)
1 (PlayBall, Runs)

0011

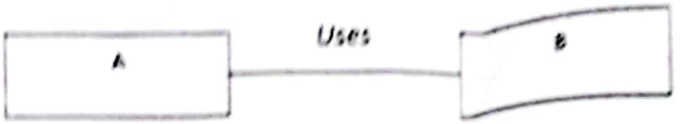
Name _____ Roll Number _____ Section _____

Question 2 (Max. Marks = 5 + 5 + 5 = 15) (CLO 2)

The following parts show partial designs (using UML 2 design class diagrams) of software applications. You are required to refactor/improve these designs using SOLID principles. Exactly one SOLID principle should be used in each part. Important instructions given in some parts must be followed to select the correct SOLID principle.

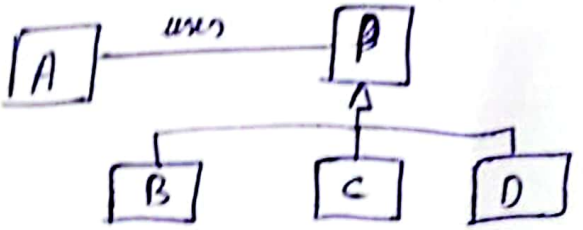
a)

Before



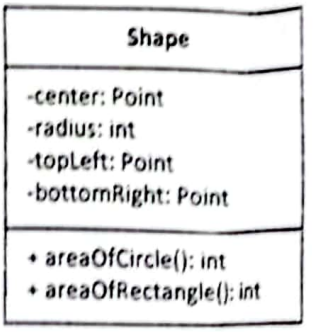
Important Instructions: In the future, Class B may be replaced by Class C or Class D.

After SOLID Principle Used: Dependency Inversion Principle

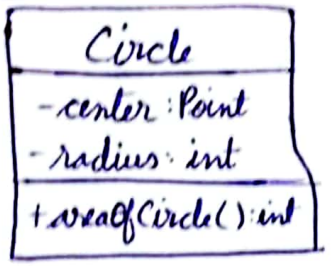


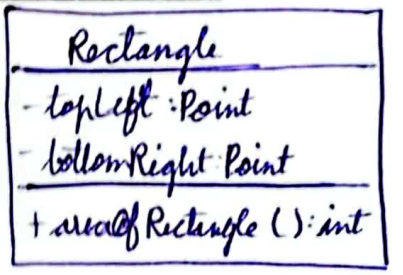
b)

Before



After SOLID Principle Used: SRP

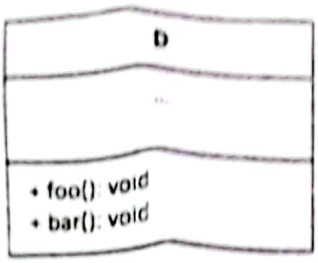




Name: _____ Roll Number: _____ Section: _____

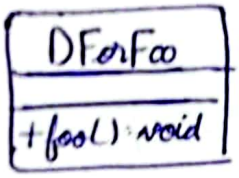
c)

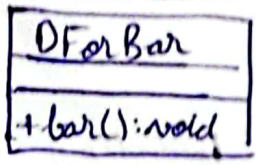
Before



Important Instructions: One client class of D uses only foo() while the other uses only bar()

After SOLID Principle Used: ISP





National University of Computer and Emerging Sciences

Lahore Campus

Software Design & Analysis (CS3004)

Sessional-II Exam

Total Time (Hrs): 1
Total Marks: 30
Total Questions: 3

Date: April 3rd 2024

Course Instructor(s)

Mr. Aamir Raheem

BCS-6A

Roll No Section Student Signature

Do not write below this line

Solve question 1 on page 1, question 2 on page 2 and question 3 on page 3. Only the first three pages will be marked!

CLO 3: Use different UML notations for software design

Q1: Consider the following class diagram:

Now give a sequence diagram to compute names of all the students registered in a given course. You can add any functions required

CLO 4: Develop software design artifacts based on requirements specifications

Q2: Write a use case description to heat food in a microwave oven

CLO 2: Implement object-oriented principles for software analysis and design

Q3: Refactor (improve) the following design using SOLID principles. Also write the name(s) of the principle(s) used.

```
class Emp {
    int id;
    char* name;
    char* email;
    int day;          // date of birth
    int mon;
    int year;
    ...
};
```