INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid-Autumn Semester Examination 2023-24

Date of Examination:_____ **Session: (FN/AN)**

Subject No :CS60059 **Subject : Object-Oriented Systems**

Department: Computer Science and Engineering

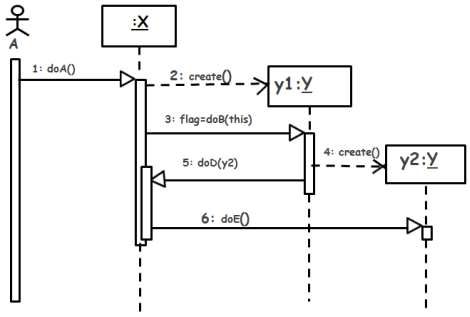
Time=2 Hours **Max Marks=100**

Important Instructions:

- Answer all questions
- **No clarification to any of the questions shall be provided during examination. In case you have any questions, you can make any suitable assumptions you feel is absolutely necessary, but please write down your assumptions clearly.**
- All answers should be brief and concise. Lengthy and irrelevant answers will be penalized.

---

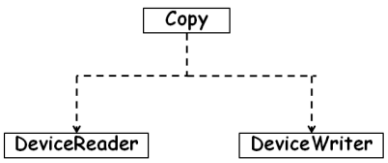
1. Consider the following sequence diagram. Write skeletal Java code for the class **Y**. Limit your code to only what is conveyed in the diagram. **[5]**



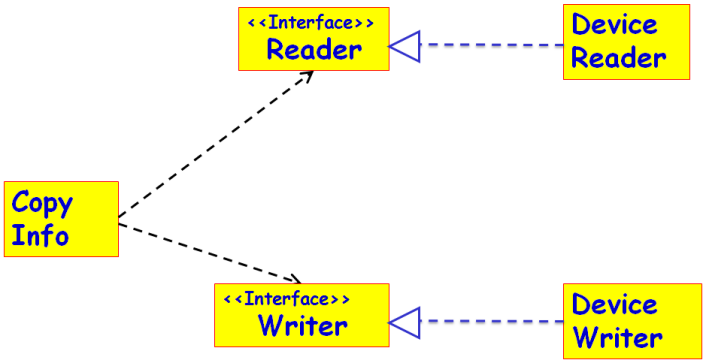
Ans:

```
public class Y {
        int flag;
        int doB(X x) {
              Y y2 = new Y();
              x.doD(y2);
              return flag;
         }
        void doE() {---}
}
```

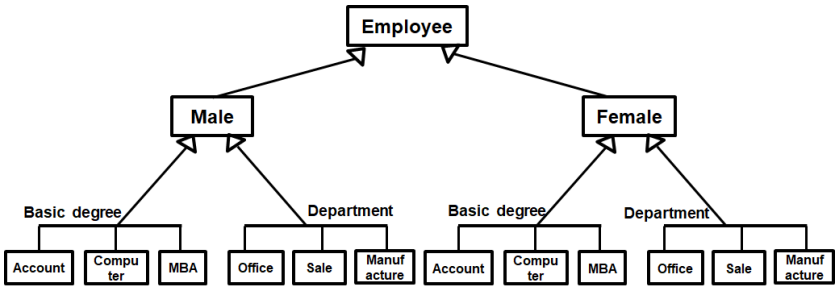
2. The following class diagram was designed by a programmer to copy information from one device to another. The copy class initializes the reader and writer device controllers and invokes the **read()** method on the **DeviceReader** and writes the read data to **DeviceWriter** by invoking its **write()** method after making necessary format changes. Name the design principle that it violates and give an improved design. **[1+4]**



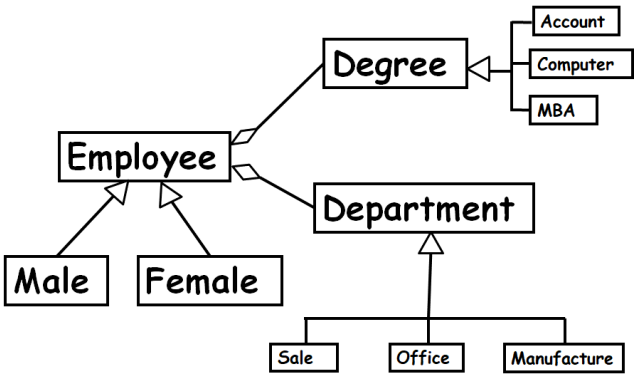
Answer: DIP



3. A company has developed an HR (Human Resources) management software to automate its HR related book keeping activities. At present the company employs persons with basic degrees in accounting, computer, and management (MBA). These employee categories have different pay structure and service conditions. The employees are posted in the following three Departments: Office, Sale, and manufacturing. However, new types of employees may be recruited and new Departments may be opened in the immediate future as the company is expanding. The programmer who was employed for developing the HR (Human Resources) software, had developed and used the following class design for the HR software.
   a) Using two brief sentences identify what is wrong with this design and the problems that it can lead to.**[2]**
   b) Suggest a suitable design (class diagram) that overcomes the identified problems. **[6]**



Ans:SRP

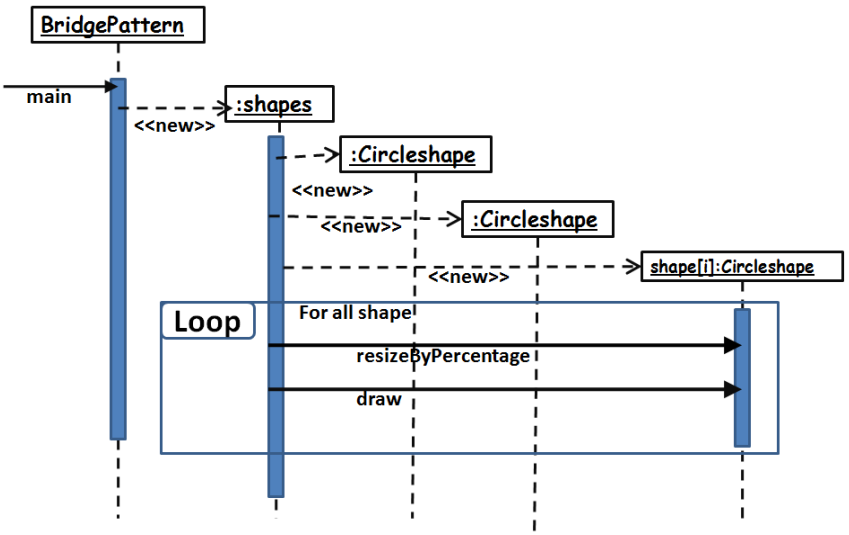


4. Develop a sequence diagram to model the interactions among various objects that take place when the **main** method of the **BridgePattern** class (whose code is given below) is invoked: **[10]**

```
class CircleShape implements Shape {
        private double x, y, radius;
        public CircleShape(double x, double y, double radius){
                this.x = x; this.y = y; this.radius = radius;
        }
        public void draw() { }
        public void resizeByPercentage(double pct) {
                radius *= pct;
         }
}
```

```
class BridgePattern {
        public static void main(String[] args) {
                Shape[] shapes = new Shape[]{
                new CircleShape(1, 2, 3), new CircleShape(5, 7, 11) }
                for (Shape shape : shapes) {
                        shape.resizeByPercentage(2.5);
                        shape.draw();
                }
        }
}
```

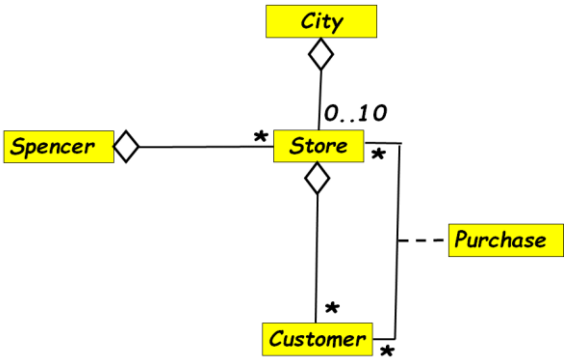
Ans:



5. Develop a UML class diagram to **model the following part requirement** for a software to be developed to automate the customer reward scheme of a shopping chain. **[10]**

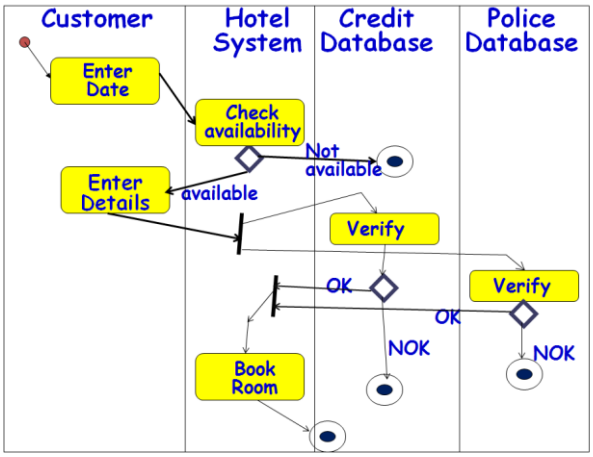
Spencer's retail has set up stores in various cities and each of these cities can have upto 10 stores. To participate in the reward scheme, a customer needs to register in any of the stores. A registered customers may buy merchandise in any of the stores. Dates of their purchase at a store and the amount of purchase need to be kept track of. Each customer accrues reward points based on his purchases.

Ans:



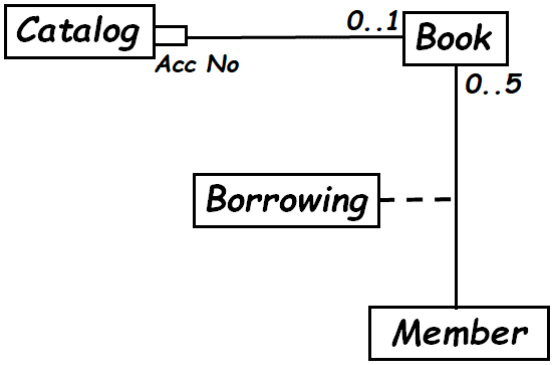
6. Develop a UML activity diagram to model the process of booking a room in a hotel as described below. **[10]**

A customer would first check whether any suitable room is available in the hotel on the desired dates. If a room is available, then the customer would enter his personal data such as telephone number, address, etc. He would also need to enter his credit card information. The system would send a request to the credit card database to authenticate the details and check whether enough credit is available to pay the hotel bills. At the same time a message would be sent out to the Police database to check if the customer is listed as a bad character. After getting favourable replies to both the queries, the system would proceed to book the room in the hotel database and display a confirmation message to the customer.



7. Consider the following part requirement of a Library Automation Software: **The library catalog would list many books. A book should be uniquely identifiable in the library catalog by its accession number. A library member at any time can borrow upto five books. A library member would be billed based on the books he/she has borrowed and the duration for which the book has been borrowed.**
   a) Draw a **UML class diagram** to model the given part requirement **[5]**
   b) Write skeletal Java code for the classes in your model. The Java code for the **Customer** class should have **addBook** and **getBook** methods for adding a **Book** to the **catalog** and getting the details of a **Book** based on its **accession number** respectively. **[7]**

Ans:

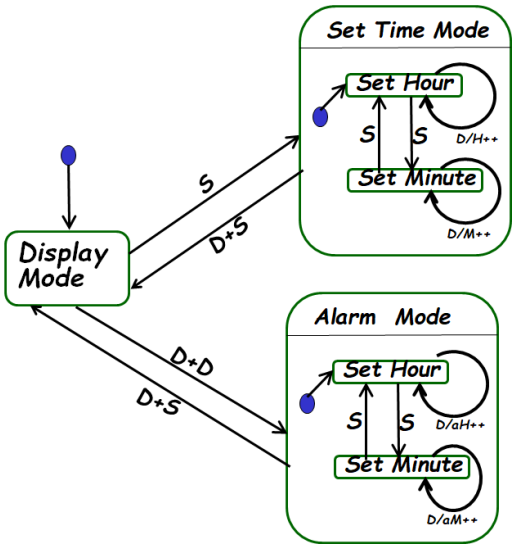



```
Class Catalog{
    Map <AccNo, Book> cat = new HashMap<AccNo,Book>();
    public Book getBook(AccNo a) {return cat.get(a);}
    public void addBook(AccNo a, Book b) {
        if(!cat.containsKey(a)) cat.put(a,b);}
}
```

8. Consider a simple digital watch having **2** buttons labelled **D** and **S**. By default, the watch keeps displaying time (**display mode**) in **hh:mm** format, and the colon keeps flashing once every second. Pressing **S** button stops the time display and takes the watch to the **set time** mode. The **set time** mode has two sub modes: **set hour** and **set minute**. In the **set time** mode, the watch starts off in the **set hour** sub mode. In the **set hour** sub mode, pressing the **D** button increments the hour display. Once the hour reading reaches **24**, it is reset to **00**. Pressing **S** toggles between the **set hour** and **set minute** modes. In the **set minute** sub mode, pressing the **D** button, advances minute display. Once the minute reading reaches **60**, it is reset to **00**. Pressing the buttons **D** and **S** together any time takes the watch to **display mode** and the watch resumes displaying the current time.

Pressing the button **D** consecutively two times would take the watch to the **set alarm** mode. The **set alarm** mode has two sub modes: **set hour** and **set minute**. In the **set hour** sub mode, pressing the **D** button increments the hour setting. Once the hour reading reaches **24**, it is reset to **00**. Pressing **S** toggles between the **set hour** and **set minute** modes. In the **set minute** sub mode, pressing the **D** button, advances minute display. Once the minute reading reaches **60**, it is reset to **00**. Pressing the buttons **D** and **S** together any time takes the watch to **display mode** and the watch resumes displaying the current time.

a) Develop the UML state machine model for the controller software of a watch whose behaviour is as described above. Use concurrent and nested states wherever required in your model. **[12]**



b) Write Java code for the controller software of the digital watch. Restrict your answer to what is implied by your state model. **[8]**

```
class clock{
    int mode=1; // 1=Display, 2=Alarm. 3=Set time
    int smode=1; // 1=Set Hour, 2=Set Minute
    int h=0,m=0; // Hour and Minute
    void S(){
            if(mode==1)  mode =3;
               if(smode==1)  smode=2;
              else smode = 1;
             }
    void D(){
               if(mode!=1) && (smode==1)) {
                        if(aH==23) aH=0;
                        else aH++;
               if(mode!=1) && (smode==2)) {
                       if(aM==23) aM=0;
                       else  aM++;
             }
  void  DD(){
            if(mode==1)  mode =2;
```

```
        }
void  DS(){
                mode =1;
        }

    }
```

9. A **Placement-assistant software** is to be developed to help automate the **placement activities** in an educational institute. The features required of this software are the following.

Placement-assistant software: The main objective is to let students apply for available placement opportunities. The employers should be able to post vacancies and be able to access the details of the student who apply against their job posting. A copy of the CV of the students who have applied, can be downloaded by the employer. Placement offers by companies and offer of acceptance by the students should also be supported on-line. While some human intervention would be needed, the process would be automated as far as possible. The following functionalities are to be supported:

i. **Enrol student:** A student desirous to participate in the placement process would fill an on-line form to provide data such as e-mail id, mobile number, and address. Also the student would have to upload a CV against a link. Once the student successfully enrols, a message is displayed for the student.

ii. **Enrol company:** A company desirous to participate in the placement process would fill an on-line form to provide data such as company name, location, turnover, contact e-mail id, mobile number, and contact address. The company would also have to pay a non refundable deposit of Rs. 50,000/. The placement officer would validate the details entered by a company. This step is intended to prevent blacklisted and weak companies from recruiting from the campus. After successful validation a enrolment id and message detailing the recruitment process is mailed to the company

iii. **Post Job:** A company can post job vacancies on-line by providing data such as job title, job requirements, eligibility criteria, closing date, and offered salary. A job posting id is generated after successful job posting.

iv. **Notify students:** The placement officer would first validate the details entered by a company. This step is intended to prevent advertisement of unrelated jobs and low quality jobs. On validation, notification mails would be sent to the students who meet the basic eligibility criteria.

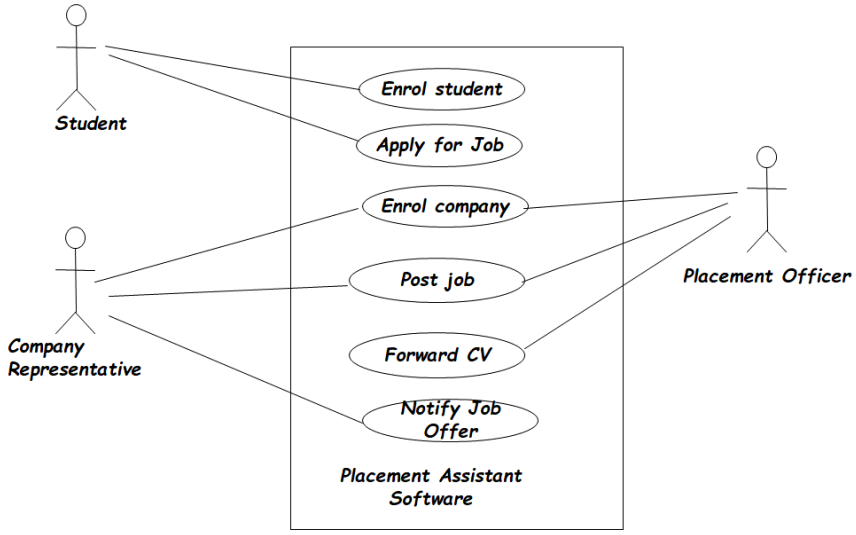
v. **Apply for a job:** A registered student can request for his/her CV to be forwarded to a company against a job posting.

vi. **Forward CV:** Based on the number of students opting for a company, the placement officer would fix a suitable threshold CGPA. The CVs of all registered students above the threshold CGPA would be forwarded to the company. The company may carry-out further shortlisting and conduct skype-based interviews for the students whose CVs were forwarded.

vii. **Notify job offer:** The company may make offers to any one or more of the students whose CVs were forwarded to it. This will cause the status of the corresponding students to change to "placed" and make them ineligible to opt for any further companies.

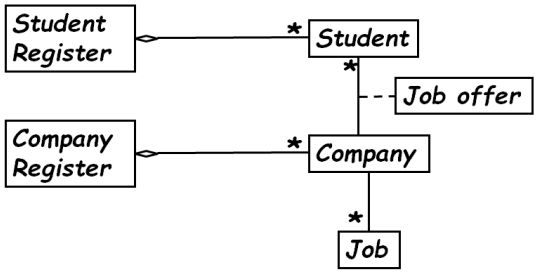
a) Develop the use case diagram. **[6]**

Ans:

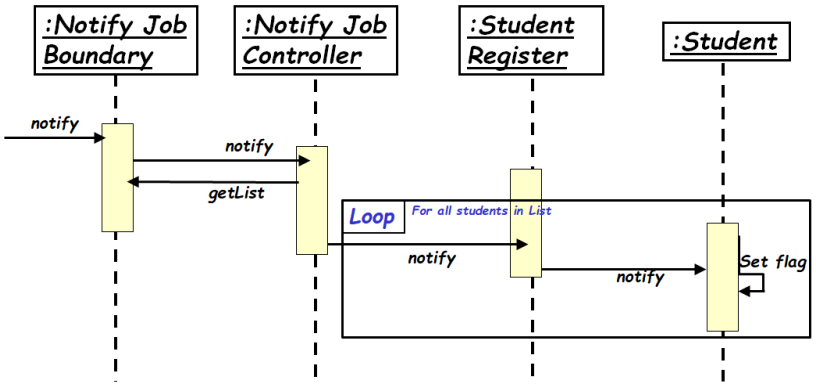


b) Develop domain model. No need to represent boundary and controller classes in your model. **[8]**

Ans:



c) Develop sequence diagram for **notify students** use case **[6]**



--- The End ---

1. It is required to develop a software to automate the process of handling research paper submissions to a conference. The following is a narrative description of the activities to be performed by the Automated Conference Paper handling (ACP).

The author completes an online form that requests the user to input author name, correspondence address, email and, title of paper. The system validates this data and, if correct, asks the author to upload the paper. The author then browses to find the correct paper on their system and uploads it. Once received and stored, the system returns to the author a reference number for the paper. The program chair allocates the papers to referees for assessment. The referees review each paper and submit to the system their decision. The program chair examines the referee reports for a paper and decides whether to accept a paper or not. After the program chair has entered his accept/reject decision for all papers, he can should be able to automatically send out review decisions to all authors through e-mail. Accepted papers are then schedule to be delivered at a conference. This involves allocating a date, time and place for the presentation of the paper.

a) Represent the user's view of the ACP softrware using a use case diagram. Write text description for one of the use cases. **[6+4]**

SAMPLE ANSWER:

b). Represent the process of organizing conferences using an activity diagram from the narrative text. **[10]**

SAMPLE ANSWER:

(1) The activity list is given below.

1) The system **requests the user to input** author name, correspondence address, email and, title of paper if the date is before the deadline.
2) The author **completes an online form** that contains the data.
3) The system **validates this data**.
4) The system **asks the author to submit** the paper, if the data is correct.
5) The author **browses their system**.
6) The author **find the correct paper** on their system.
7) The author **submits the paper**.
8) The system **receives the paper**.
9) The system **stores the paper**.
10) The system **assigns a reference number** for the paper.
11) The system **returns the reference number** to the author.
12) The author goes back to activity 4) if they want to submit another paper.
13) System **allocates papers to referees** for assessment.
14) Referee **reviews each paper**.
15) Referee **submits their assessments** to the system.
16) The programme organiser **make decisions** on each paper's acceptance.
17) The system **emails the author** of each paper of the decision.
18) The system **creates a schedule** for delivering accepted papers at a conference by allocating a date, time and place for the presentation of each paper.

(2) The action/activity nodes are marked on the activity list as bold font.

(3) The actors are the name part of the activities.

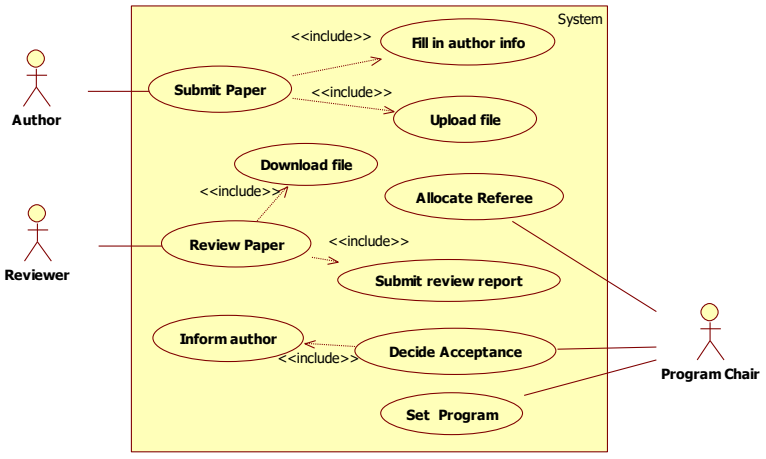
(4) In this exercise, the actions are all sequentially ordered as in the activity list. The parallelism and concurrency only occur in the form of different referees can perform their reviews and assessments in parallel.

(5) The actors are:

(a) Author; (b) The system; (c) Referee; (d) Conference programme organiser;

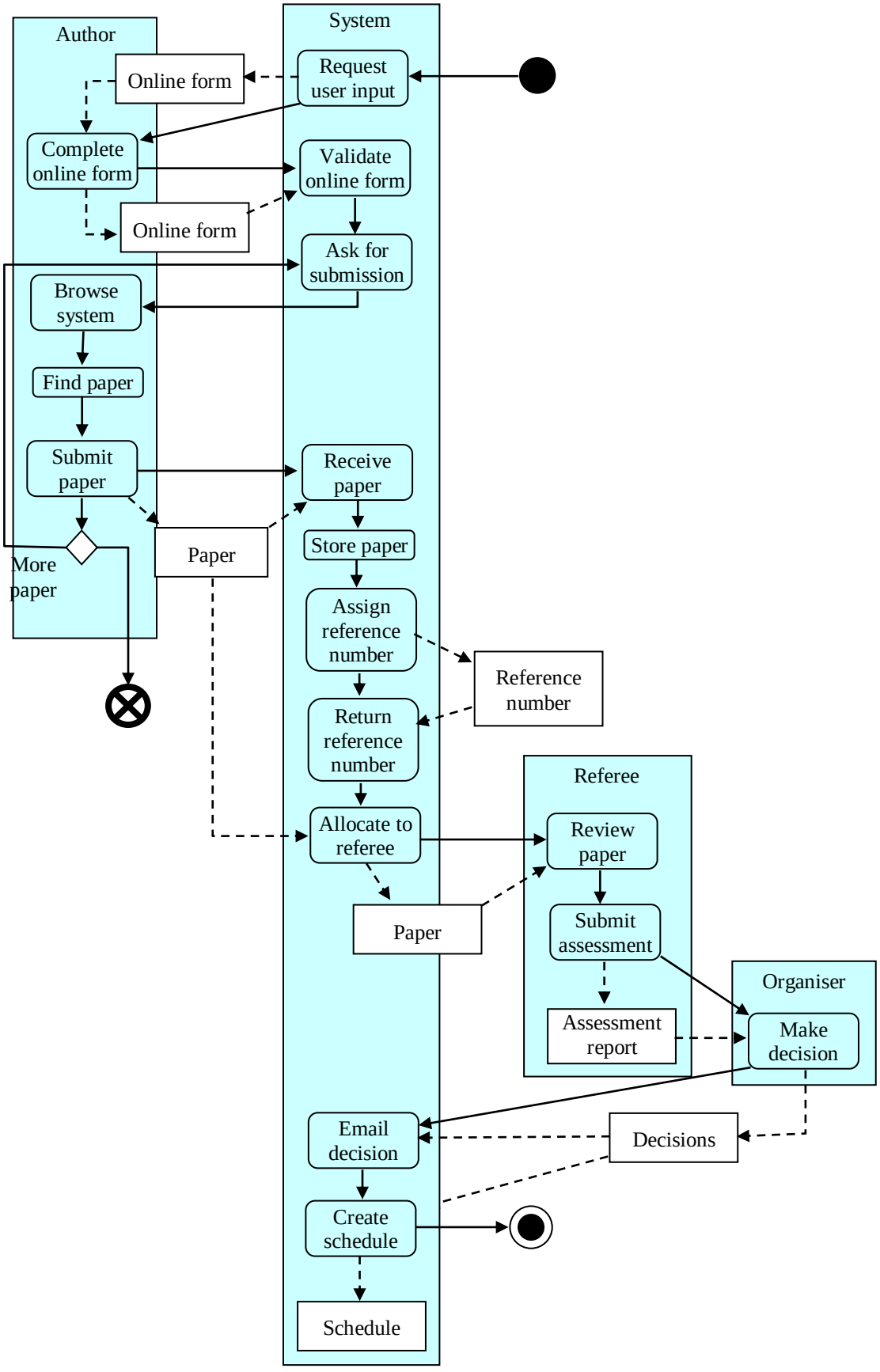
(6) The objects involved in this process are:

(a) online form; (b) paper; (c) paper's reference number; (d) assessment report;



(e) decision; (f) conference schudel.

The derived activity diagram is given below.


Author
System
Online form
Request user input
Complete online form
Validate online form
Online form
Ask for submission
Browse system
Find paper
Submit paper
Receive paper
Paper
Store paper
More paper
Assign reference number
Reference number
Return reference number
Referee
Allocate to referee
Review paper
Paper
Submit assessment
Organiser
Assessment report
Make decision
Email decision
Decisions
Create schedule
Schedule

10. Krishna air travels (KAT) allows its registered customers to book cars for travel to outstation locations or for dropping at the local Railway station. Before any one can book a car, he/she is required to register once with KAT by providing name, residential address, phone number, and identity proof. KAT entertains registration requests from only the students and faculty members of IIT and rejects requests received from any other person. Once a request from a registered student or faculty is received by the KAT reception, the request is sent to the accounts department which prepares a duty slip (indicates driver name, Vehicle number, customer name and address, and the date, time, and other specifics of the travel) and hands it out to the driver. Whenever, outstation travel is involved, in addition to duty slip generation, the accounts Department also sends the vehicle to the maintenance department to make it journey worthy and to fill up diesel. After the journey is performed, the driver takes the signature of the customer on the duty slip and hands it over to the accounts department. The accounts department prepares the bill that the customer can download online. After the customer pays the bill online, the records of journey are archived by the accounts department.
    a) Develop use case diagram. No text description is necessary. **[8]**

    b) Develop activity diagram for car booking, including the customer registration process. **[10]**

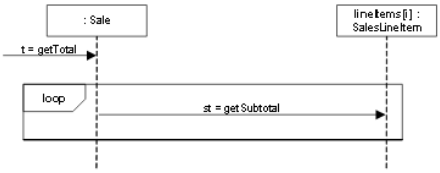
    c) Develop a class diagram to show the classes and their relationships from the text description (called the preliminary class diagram or domain model) **[10]**

11. Develop the state machine model of the behaviour of the controller software for a laptop described below. Give skeletal Java or C code implementation of the laptop controller. **[15+10]**

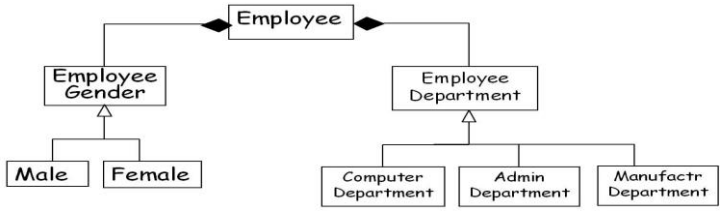
    A laptop is in the off mode to start with. It becomes ON by pressing the power switch. While in the power ON mode, if the power switch is pressed, it enters the off mode. The behaviour of the laptop in the ON mode is described in the following: When AC mains power is available, the laptop works in the normal operating mode provided the user regularly interacts with it. When the user remains inactive for a minute, the laptop enters a low power *screen saver mode*. If the user remains inactive for further two minutes, it enters into a power saving *blank screen* mode. In the screen-saver and blank screen modes, user pressing any key brings the laptop to the normal operating mode. After five minutes of user inactivity in the blank screen mode, the laptop enters a very low power *sleep mode*. On the other hand, if the laptop is on battery power, the normal operating mode is a *dimmed screen* mode to save power. The laptop enters a power saving *blank screen* mode if the user is inactive for a minute. In this mode, the user pressing any key brings it to the normal operating mode. The laptop enters a very low power sleep mode if the user remains inactive in the blank screen mode for three minutes. Whenever the laptop is in the sleep mode, depending on whether the laptop is on battery or AC power, the user can get back to the corresponding normal working mode by pressing the power off button. In any mode other than the sleep and off modes, the power off button starts a shutdown procedure. Also, when the laptop is working on battery and the battery power falls below 5% of the maximum energy value, an automatic shutdown procedure is started.

12. Develop a sequence diagram each to model the interactions implemented in the following code segments: **[5+5]**

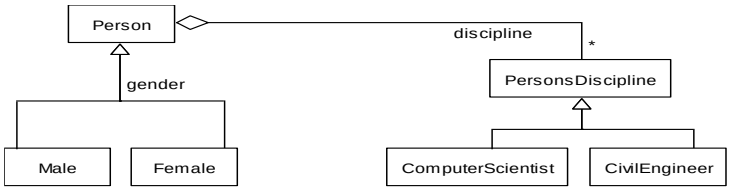
    a)
```
public class Sale  {
    private List<SalesLineItem> lineItems = new ArrayList<SalesLineItem>();
    public Money getTotal()  {
        Money total = new Money();
        Money subtotal = null;
        for ( SalesLineItem lineItem : lineItems ) {
            subtotal = lineItem.getSubtotal();
             total.add( subtotal );
        }
        return total; }
// . .  }
```



13. Consider the following class design for the HR (human resource) application for an organization. According to the organization's HR policy, an employee may be transferred among any of its different Departments. What is wrong with the class design given below? Give an improved class diagram. **[1+4]**



Ans

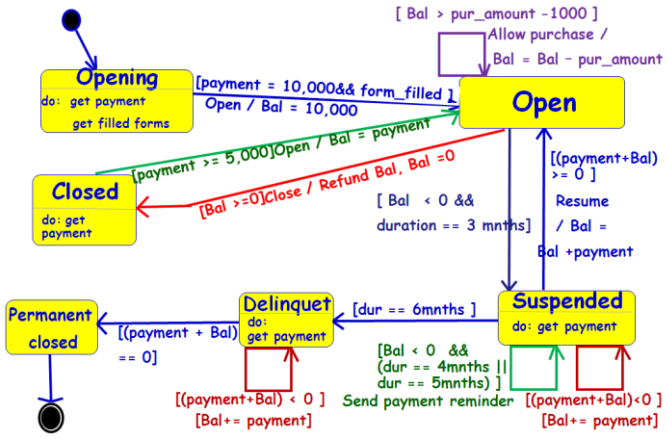


14. Consider the following informal requirements that has been gathered for the development of an e-commerce application:

A "customer care" application is used by Customer Service Representatives (CSRs) who talk to customers over the telephone. A given call consists of a conversation between a CSR and a customer, and results in the creation of "service requests", such as: query goods, purchase goods, and several account-related requests such as open, close, deposit money, receive payment, reopen, or resume an account. An account can be opened by a customer depositing Rs. 10,000/- and filling up the necessary forms on-line; on completion of these two steps an account number is assigned to the customer and the balance in the account is made Rs. 10,000/-. Purchases through telephone calls to CSRs are allowed only on open accounts, subject to the restriction that the account balance must exceed the amount of purchase or at most fall short by Rs. 1000/-. Whenever an account close request is made, the balance amount is refunded to the customer by transferring this sum to the customer's bank account. An account cannot be closed if there is money owed on it; in which case, if sufficient amount of money is deposited into it so that no money is no longer owed on it, the account can be closed. A closed account can be reopened by paying at least Rs. 5,000 into the account and issuing a reopen request. A suspended account may have service resumed by a resume request, only if there is no money owed on it. An account will automatically get suspended if money is owed and the required amount has not been deposited for 3 months, in which case a payment reminder letter will also be sent. Payment reminder letters are sent at the end of the 4th & 5th months, if the required money has not been deposited by the customer by that time. At the end of 6 months, the account becomes delinquent, and may never be reopened, but of course the customer can pay his or her debt to permanently close the account.

(a) Develop a state model for the account class**. [10 Marks]**

(b) Write skeletal Java code that can be inferred from the Account class state model that you have developed for the part (a) of this question. **[5 Marks]**




```
enum State {Opening, Open, Closed, Suspended, Delinquet, Permanent_closed};
enum Event {open, purchase, rcvPayment, idleTimer, close};
```

```
static State s = opening;
Event event = open;
while(true){
   switch(s){
            case `Opening' : getPayment();   getFormsFilled();
                                    if(event == open && payment==10,000 && forms_filled ==y)
                                         s = Open;
                                     break;
       case `Open'      :  switch(event) {
                                    case `purchase' : if(Bal >= (purchase-amount-1000)){
                                                            purchase(); Bal = Bal – purchase_amount;
                                                             }break;
                                    case `close' : if(Bal >=0) {
                                                          if(Bal>0)
                                                           if(refund()){
                                                                 Bal =0; s = Closed; }
                                                          }break;
case `idleTimer' : if( Bal < 0 && duration== 3){
                                                             s = Suspended;
                                                             }
                                                            break;
                                    }  break;
       case `Suspended' : switch(event) {
                                        case `rcvPayment' : Bal += payment;
                                                                 if(Bal>= 0) s = Open;
                                                                 break;
                                          case `idleTimer' : if(duration == 4 || duration==5)
                                                                   paymentReminder();
                                                                   if(duration==6){ s = Delinquet; }
                                                                    break;
                                        }break;
       case `Delinquet' :      if(event=rcvPayment) {
                                          Bal+=payment;
                                          if(Bal>=0) s= Permanent_closed;}
                                        break;
case `Closed' : if(event=rcvPayment ) {
                                   Bal+=payment;
                                   if(Bal>=5,000) s= Open;}
                                   }break;
            }
}
```

15. IIT students' Hall Management Center (HMC) has requested us to develop a software to automate various book-keeping activities associated with its day to day operations. From the following description of the requirements for the HMC software project, develop the use case model, domain model, and the sequence diagram for the compute hostel dues use case. **[8+8+5]**

After a student takes admission, he/she would present a note from the admission unit to the clerk at HMC, along with a form filled with his/her name, permanent address, contact telephone number, and a photograph. He/she is then allotted a hall, and also a specific room number. A letter indicating the allotted room should be printed out to be given to the concerned student.

Students incur mess charges every month depending on the number of meals that they consume over a month. The mess manager would input to the software the total amount chargeable to each student in a month on mess account.

Each room has a fixed room rent, but the room rent depends on the exact room type and the Hall. The rooms are either single-seated or twin-sharing. The newly constructed halls have higher rent compared to some of the older halls. Twin sharing rooms have lower rent. The HMC chairman enters the room rents for the different types of rooms. He should also be able to revise the room rents whenever required.

Each hall provides certain amenities to the students such as reading rooms, play rooms, TV room, etc. A fixed amount is levied on each student on this count. This amount is estimated and entered by the HMC chairman at the beginning of an academic year.

The estimated due from a student towards the residence and mess charges for a semester is computed at the start of a semester, which the student can see using a web browser. At the end of a semester, the difference between the actual bill amount for the semester and the paid amount is computed. The balance amount (+ve or –Ve) is adjusted in the next semester's dues. The total amount collected from each student of a hall towards mess charges, and the due amount is handed over to the concerned mess manager every month. For this, the computer needs to print a sheet indicating the total amount due to each mess manager. Printed cheques are issued to each manager and their signatures are obtained on the receipt sheet.

The HMC receives an annual grant from the Institute for upkeep of the halls, gardens, and providing certain general amenities to the students. The HMC chairman should be provided with a functionality that would support distribution of the grant among the different halls and cheques should be printed based on the grants made to the halls. The Wardens of different halls should be able to enter their expenditure details against this allocation.

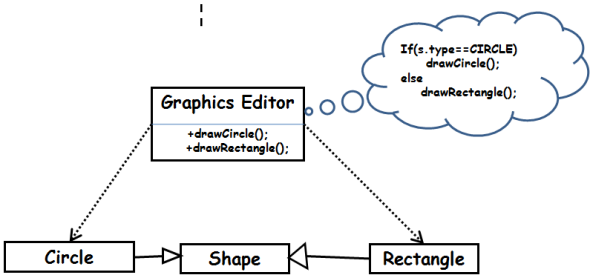
The HMC incurs petty expenses such as repair works carried out, stationery, etc. It should be possible to enter these expenses and it should be debited from its yearly grant. The HMC chairman should be able to view at any time the statement of accounts. The HMC chairman would take a print out of the annual consolidated statement of accounts, sign and submit it to the Institute administration for approval and audit verification.

Consider that you have undertaken to incrementally develop a business application that incorporates a graphics Editor. The code segment of the first version of your graphics Editor is the following. Draw the class diagram for the code. Identify the reasons as

to why it cannot be considered to be a good design. Suggest a suitable class diagram that overcomes its shortcomings. Give the skeletal Java code corresponding to your improved design. **[2+2+3+3]**

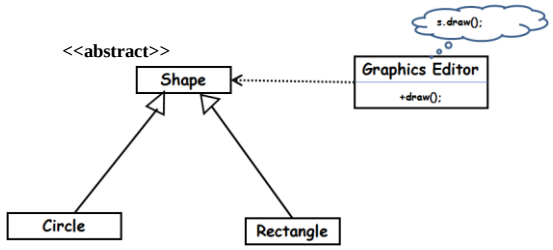
```
class GraphicEditor {
        public void drawShape(Shape s) {
                if (s.m_type==1)
                        drawRectangle(s);
                else if (s.m_type==2)
                        drawCircle(s);
        }
        public void drawCircle(Circle r) {....}
        public void drawRectangle(Rectangle r) {....}
}
```

```
class Shape {
        int m_type;
}
 class Rectangle extends Shape {
Rectangle() {
                super.m_type=1;}
}
 class Circle extends Shape {
        Circle() {
                super.m_type=2;}
}
```



Answer:

a) Violates OCP



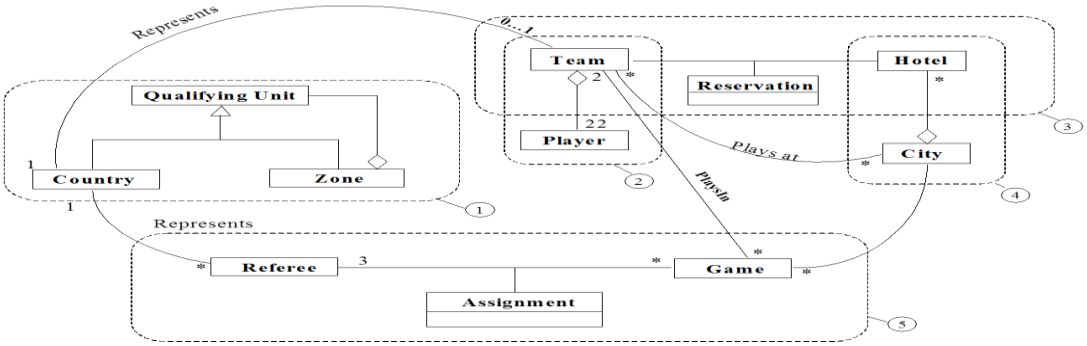

```
class GraphicEditor {
        public void drawShape(Shape s) {
                s.draw();
        }
}
```

```
class Shape {
        abstract void draw();
}

class Rectangle extends Shape  {
        public void draw() {
                // draw the rectangle
        }
}
```

```
class Shape {
        abstract void draw();
}

class Rectangle extends Shape  {
        public void draw() {
                // draw the rectangle
        }
}
```

Consider the following excerpt from the SRS document for a software being developed to automate the book-keeping activities associated with arranging the logistics for the conduction of the 2018 FIFA World Cup Football tournament to be played in Russia. In this tournament, teams represent countries and each team is made up of upto 22 players. The games are to be conducted in a number of selected cities. Each team should play at most two games in any specific city. The set of referees would be drawn such that there is at least one referee each qualifying country. Two Referees are to be assigned to each game. Reservations are to be made for the referees and the players for the scheduled days in one or more hotels in the city where the teams are scheduled to play.

Identify the domain classes and the identifiable relations among them and represent those in a UML class diagram. **[5]**



Consider the following informal requirements that have been gathered for an e-commerce application development:

A "customer care" application is used by Customer Service Representatives (CSRs) who talk to customers over the telephone. A given call consists of a conversation between a CSR and an authorized contact person representing a customer account, and may involve the CSR creating "service requests", such as: query goods, purchase goods, and several account-related requests such as open, close, deposit money, receive payment, reopen, or resume an account. An account can be opened by a customer depositing Rs. 10,000/-, filling up the necessary forms on-line; and then calling the CSR. Once the CSR gives an "open" request, an account number is assigned to the customer and the Rs. 10,000/- paid by the customer is credited to the account. Purchases through telephone calls to CSRs are allowed only on open accounts, subject to the restriction that the account balance must exceed the amount of purchase or at most fall short by Rs. 1000/-. Whenever an account is closed, the balance amount is refunded to the customer by transferring this sum to the customer's bank account. An account cannot be closed if there is money owed on it; in which case, if sufficient amount of money is deposited into it so that no money is any longer owed on it, the account can be closed. A closed account can be reopened by paying at least Rs. 5,000 into the account and issuing a reopen request. A suspended account may have service resumed by a resume request, only if there is no money owed on it. An account will automatically get suspended if money is owed on it and the required amount has not been deposited for 3 months, in which case a payment reminder letter will also be sent. After account suspension, payment reminder letters are sent to the customer at the end of the 4th and 5th months, if the required money has not been deposited by that time. At the end of 6 months, the account becomes delinquent, and may never be reopened, but of course the customer can pay his/her debt to permanently close the account.

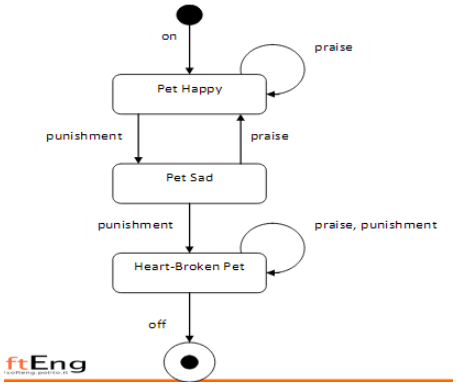
1. Develop a state model for the account class**. [10 Marks]**

2. Write skeletal Java or C code to implement the Account class state model that you have developed for the part (a) of this question. **[10 Marks]**

16. Develop the Interaction diagram for the Use Case **Add Appointment of a Calendar application**: The execution of the use case begins when the user chooses to add a new appointment by clicking on a date in the Calendar UI. The UI notices which part of the calendar is active and pops up a window for that date and time. The user enters the necessary information in this pop-up window about the appointment's name, location, start and end times. The Appointment window does not accept any appointment that has invalid information, such as an empty name or negative duration. The Calendar checks if the new appointment would conflict with any of the user's already booked appointments. If there are no conflict with any of the user's appointments, the calendar records the new appointment in the user's list of appointments.
Alternate scenario 1: If the user already has an appointment at the same time at which the user is trying to add an appointment, the user is shown a warning message and asked to choose any other available time. The appointment is booked based on the user's response without any further checking and interactions. **[10 marks]**

2. Develop the UML state machine model of a digital pet software. The behavior of the digital pet described in the following. Also, translate your state machine model into skeletal Java code. **[8+7]**

When the program is started, the pet starts out happy. When the pet is happy and it receives punishment, it becomes sad. If the pet is happy and receives either praise or food, it remains happy. If the pet is sad and it receives either food or praise, it becomes happy. If the pet is sad and receives punishment, it becomes heart broken and giving it any one of food, praise or punishment, has no effect. However, when the pet is heartbroken and is given both praise and food together, it becomes happy.



17.

a) Consider the following code segment. Develop a sequence diagram to model the interactions that occur when the method **getTotal** is invoked. **[5]**

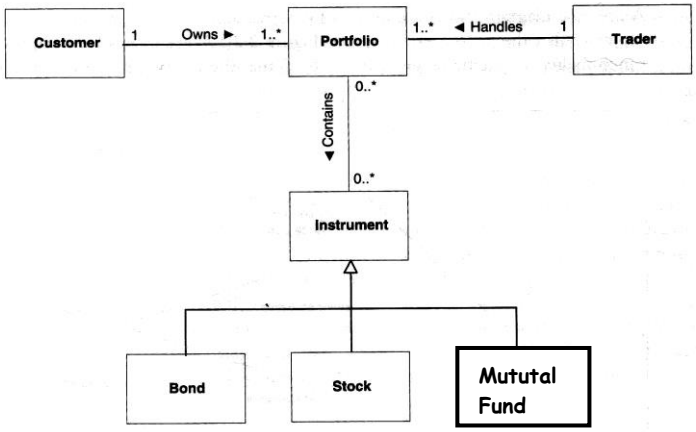
```
public class Sale  {
private List<SalesLineItem> lineItems = new ArrayList<SalesLineItem>();
public Money getTotal()  {
        Money total = new Money();
        Money subtotal = null;
        for ( SalesLineItem lineItem : lineItems ) {
                subtotal = lineItem.getSubtotal();
                 total.add( subtotal );
        }
        return total; }
// . .  }
```

Develop the UML state machine model for the controller software of a wrist watch, whose behavior is described below. Use of concurrent states wherever required is recommended. **[10]**

A digital wrist watch has 3 buttons labeled M, S, and A. By default, the watch keeps displaying time in an analog mode using hour, min, and second hands. In this mode, pressing the S button changes the time display to digital form. Pressing S button in the digital display mode changes the display back to analog display mode, and so on. That is, S button toggles between digital and analog display modes. In both analog and digital display modes, pressing the button A has no effect. However, in any of analog or digital display modes pressing the button M once takes the watch to hour mode. In the hour mode, the time display is in digital form with the hour reading blinking, and pressing the button A advances the hour reading using a 24 hour format. In the hour mode, pressing the button M once takes the watch to minute mode. In the minute mode, the minute reading blinks and pressing the button A advances the minute reading. Pressing the button M in the minute mode, takes the watch to digital display mode and the watch resumes displaying the current time in digital form without any blinking.

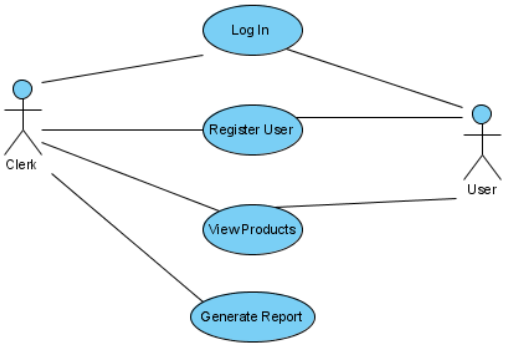
Develop a class diagram for the following problem.**[10]**

A certain finance company specializes in handling client portfolios. The company employs many finance professionals (designated financé wizards) specialized in client portfolio management. A client may own multiple portfolios. A client can have various types of financial instruments in his portfolio. At present, three main types of financial instruments are supported, bonds, stocks, and mutual fund units. A finance wizard may manage one or more portfolios.



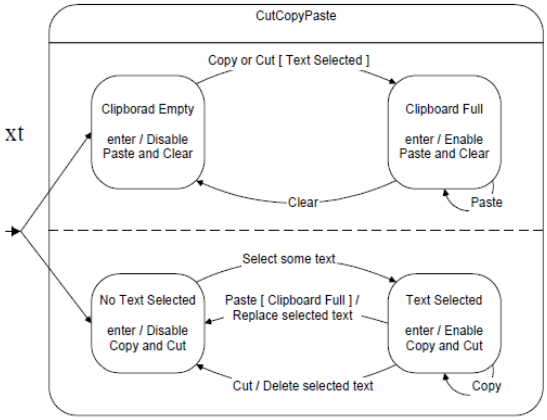
Q.4 Develop the use case diagram for the following web-based Inventory Management System (IMS). No text description is necessary. **[5]**

Using IMS, a clerk should be able to view online the specific products that are in stock. IMS should allow the clerk to generate an invoice for the goods for which he observes the stock to be below a threshold. To generate an invoice, the clerk must first log in. If a clerk is a first time user, he must first create a login account. When a sales occurs using the sales management software (SMS), a reduce inventory function of the IMS is automatically invokes. When supply arrives, the clerk should be able to update the inventory. The clerk can generate a report summarizing the daily transactions. For certain features of the SMS software, it would require IMS to report the inventory level of any required goods.



Q.5 Develop the UML state machine model for the following text editor. Give the Java code that can be generated from your state machine diagram. **[10+5]**

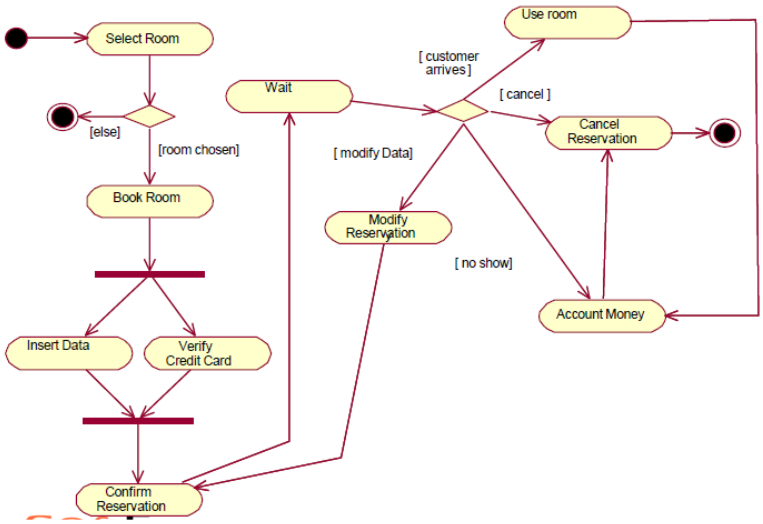
A text editor should support cut, copy, and paste functionalities. Paste only works if there is text in the clipboard. As the result of a paste command, text in the clipboard is pasted. The clip board may only be filled with a copy or cut command. Cut and copy commands only work if there is text selected. Cut deletes the previously selected text. The clipboard may be emptied by pressing CTRL-C.



Q.6 Develop the UML activity diagram for the process of booking a room in a hotel as described below. **[10]**

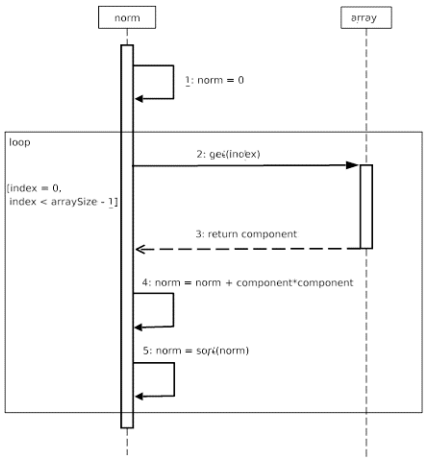
When customer needs a room on specific dates, he first checks whether a room is available in the hotel on those dates. If room is available, then he enters the data such as telephone number address, etc. He also enters the credit card information. The system sends a request to the credit card database to check whether enough credit is available to pay the hotel bills. At the same time a message is sent out to the Police database to check if the customer is listed as a bad character. After getting favourable

replies to both the queries, it proceeds to book the room in the hotel database. The Hotel information system allows customer to modify or cancel his/her reservation. If customer arrives he uses the room and then he pays, in case of no-show a penalty is charged to his credit card and the booking for the subsequent days is cancelled.



Q.4 Develop the sequence diagram for the following Java code. [10]

```
Class Array {
...
public:
// return the index-th component of the array
double get(int index);
...
};
double norm(const Array& myArray) {
double theNorm = 0;
for(int index = 0; index < myArray.size() - 1; index++) {
theNorm = theNorm + myArray.get(index);
}
theNorm = sqrt(theNorm);
return theNorm;
```



Q.6 Design the Sequence diagram for our Courseware Management System case study application. Because a Sequence diagram represents the dynamic flows in an application, we will aim to represent one of the flows using a Sequence diagram. In a previous analysis, the following use cases for the Courseware Management System were defined:

- View courses

- Manage topics for a course
- Manage course information
- View course calendar
- View tutors
- Manage tutor information
- Assign courses to tutors

For these use cases, in a previous phase, classes and interfaces have been modeled using the class diagram. The next step will be to combine the flow defined by the use cases and the classes involved in the use cases together to represent the different flows in the Courseware Management System.

Represent the "Manage course information" flow using a Sequence diagram knowing that this flow contains one participant: the Course Administrator.

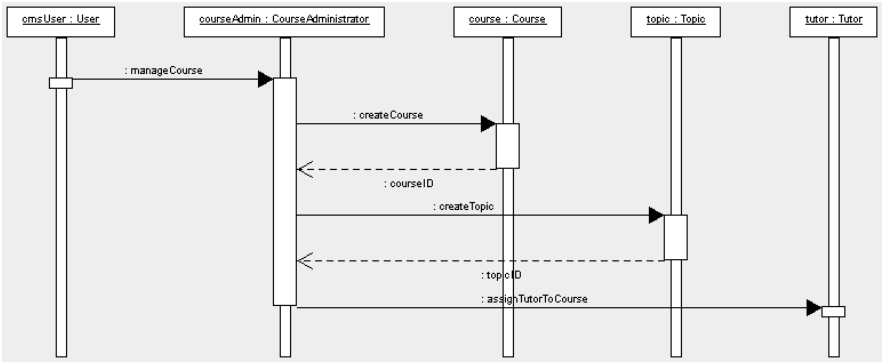
Apart from this, there are a few entities with which the course administrator interacts in this flow—Course, Topic, and Tutor.

Solution

The sequence of steps carried out in the "Manage course information" flow are:

- A user who is a course administrator invokes the manage course functionality.
- The manage course functionality of the course administrator invokes either the course creation or course modification functionality of a course.
- After the course is either created or modified, the manage topic functionality of the course administrator calls the topic creation or modification functionality of a topic.
- Finally, the user invokes the assign tutor to course functionality of the course administrator to assign a tutor to the selected course.

Now, let us model these steps into a Sequence diagram for the "Manage course information" functionality.



1. Consider the following Java code segments. Draw a sequence diagram to represent the message exchanges that take place among objects when the printGameMoney() method of the Game class is invoked. **[4]**

```
class Ticket {
    int gameID;
    double price;
    int seatnumber;
    public int getPriceOfTick();
}
```

```
Class Game{
  private List <Ticket> tickets=new Arraylist<Ticket>();
  public static int printGameMoney(){
     double total = 0;
      for (final Ticket ticket : tickets)   {
         final double tickMoney = ticket.getPriceOfTick();
         total = total + tickMoney;
     }
  }
}
```

18. A set of flights operate from each airport. Each flight is identified by its flight number. Each flight is characterized by the type of the aircraft manufacturer (e.g. Boeing or Airbus), aircraft model (e.g. 737, 747, 787 etc.), an origin airport, a destination airport, scheduled times at the origin and the destination, days of the week in which it operates, and the total number of seats in the flight. A passenger can book on a flight for a specific date. Each passenger has a name, a mobile number, and address. A flight booking is characterized by a transaction number, ticket Price, and any special amenities booked (such as preordered meals). Draw a UML class diagram to model this. For the different classes, write skeletal Java code that can be inferred from the information present in your design. **[10+10]**

19. Develop the state machine model of the behaviour of the controller software for a laptop described below. Give skeletal Java or C code implemention of the laptop controller. **[15+10]**

    A laptop is in the off mode to start with. It becomes ON by pressing the power switch. While in the power ON mode, if the power switch is pressed, it enters the off mode. The behaviour of the laptop in the ON mode is described in the following: When AC mains power is available, the laptop works in the normal operating mode provided the user regularly interacts with it. When the user remains inactive for a minute, the laptop enters a low power *screen saver mode*. If the user remains inactive for further two minutes, it enters into a power saving *blank screen* mode. In the screen-saver and blank screen modes, user pressing any key brings the laptop to the normal operating mode. After five minutes of user inactivity in the blank screen mode, the laptop enters a very low power *sleep mode*. On the other hand, if the laptop is on battery power, the normal operating mode is a *dimmed screen* mode to save power. The laptop enters a power saving *blank screen* mode if the user is inactive for a minute. In this mode, the user pressing any key brings it to the normal operating mode. The laptop enters a very low power sleep mode if the user remains inactive in the blank screen mode for three minutes. Whenever the laptop is in the sleep mode, depending on whether the laptop is on battery or AC power, the user can get back to the corresponding normal working mode by pressing the power off button. In any mode other than the sleep and off modes, the power off button starts a shutdown procedure. Also, when the laptop is working on battery and the battery power falls below 5% of the maximum energy value, an automatic shutdown procedure is started.

Develop the UML state machine model of a digital pet software described in the following and write the corresponding skeletal Java code. **[8+7]**

When the program is started, the pet starts out happy. When the pet is happy and receives either praise or food, it remains happy. If the pet is happy and it receives punishment, it becomes sad. If the pet is sad and it receives either food or praise, it becomes happy. If the pet is sad and receives punishment, it becomes heart broken and giving it any food, praise or punishment, has no effect. However, when the pet is heartbroken and is given both praise and food together, it becomes happy. Also, anytime it is not happy and asked out to play, it becomes happy.

20. A real-estate web application is used by ordinary users looking to buy a property, and also by the real estate agents, and managers of the company hosting the application. To be able to invoke any functionalities of the software, any user must first login to the system. If a user does not have any login account, he can create one by filling up the necessary information online. Once created, the accounts of ordinary users becomes active instantly. However, the login accounts of the managers and agents need to be first approved by the administrator, before they can become active. An agent can post into the software the details of a property that is available for sale. Any user can view the listing of various properties available for sale. An agent can update the posting for an available property, for example when a property is sold or the terms and conditions are altered. However, any update to a posting becomes visible only after a manager approves it. Managers can remove a listing when a property is sold or the company ceases to market the property for any reason. *Based on this description, draw a use case diagram. No text description is necessary*. **[10 marks]**

Consider the following informal requirements that has been gathered for the development of an e-commerce application:

A customer can on-line create "service requests", such as: query goods, purchase goods, and several account-related requests such as open, close, deposit money, receive payment, reopen, or resume an account. An account can be opened by a customer depositing Rs. 10,000/- and filling up the necessary forms on-line; on completion of these two steps an account number is assigned to the customer and the balance in the account is made Rs. 10,000/-. Purchases requests are allowed only on open accounts, subject to the restriction that the account balance must exceed the amount of purchase or at most fall short by Rs. 1000/-. Whenever an account close request is made, the balance amount is refunded to the customer by transferring this sum to the customer's bank account. An account cannot be closed if there is money owed on it; in which case, if sufficient amount of money is deposited into it so that no money is no longer owed on it, the account can be closed. A closed account can be reopened by paying at least Rs. 5,000 into the account and

issuing a reopen request. A suspended account may have service resumed by a resume request, only if there is no money owed on it. An account will automatically get suspended if money is owed and the required amount has not been deposited for 3 months, in which case a payment reminder letter will also be sent. Payment reminder letters are sent at the end of the 4th & 5th months, if the required money has not been deposited by the customer by that time. At the end of 6 months, the account becomes delinquent, and may never be reopened, but of course the customer can pay his or her debt to permanently close the account.

(c) Develop a state model for the account class**. [10 Marks]**

(d) Write skeletal Java code that can be inferred from the Account class state model that you have developed for the part (a) of this question. **[5 Marks]**