



**UNIVERSITY OF COLOMBO, SRI LANKA**



**UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING**

**DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)**  
**Academic Year 2004/2005 – 2nd Year Examination – Semester 3**

***IT3102: Object Oriented Systems Development***  
***PART 2 – Structured Question Paper***

**5<sup>th</sup> March, 2005**  
**(ONE AND HALF HOURS)**

**To be completed by the candidate**

BIT Examination Index No: \_\_\_\_\_

**Important Instructions:**

- The duration of the paper is **1 ½ (one and half) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **13 pages**.
- **Answer question 1 and any 2 of the other questions only.**
- All questions will carry equal marks.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.  
If a page is not printed, please inform the supervisor immediately.
- **Non-programmable Calculators may be used.**

**Questions Answered**

Indicate by a cross (x), (e.g. ☐) the numbers of the questions answered.

To be completed by the candidate by marking a cross (x).	Question numbers			
	1	2	3	4
To be completed by the examiners:				

Based on the following case study answer question 1.

1)

## **InfoTech Agency**

### **Case Study**

The following text describes the operations of the InfoTech Agency.

InfoTech Agency is a famous bookshop which sells technical books, CDs, tapes and videos imported directly from international publishers. It also manages a library of those items at a special 3-tiered subscription rate for its customers. Library members receive special discounts whenever they purchase any item from InfoTech Agency. Operations of the library and bookshop are integrated together.

The library is set up along conventional lines. There are restrictions on the number of items which can be borrowed depending on the level of subscription. Levels A,B and C are allowed 3,2 and 1 items respectively. Library items should be returned within seven days but they can be extended for another week if there is no reservation by another member of the library.

Library items include books, CDs, tapes and videos. They can be categorized as permanent reference items (PR) or lending items. PR items can be used inside the library but are not allowed to be borrowed. There can be more than one copy available for any item.

In order to borrow a library item or to buy, a customer selects it and takes it to the entrance counter which handles both lending and selling operations. However, there is a separate counter for the cashier. Library members have a special card with a photo of the member. The person at the counter first checks the library card for membership status. Subsequently he will check for overdue books and whether he has already reached the borrowing limit. If there is no problem, then the member can leave with his loan materials and that information will be recorded. On the other hand, if a person wants to buy some items, the clerk issues an invoice to the customer and the relevant amount should be paid to the cashier before items are issued. The customer can claim a special discount, if he posses a library card of the InfoTech Agency library.

When a member returns borrowed items, the clerk checks whether the items are reserved items. If so, he will take necessary steps to inform the person who has reserved it. If there is a fine to be paid due to damage or items being overdue, the member should settle it to reactivate his membership status.

By giving details of library items and the membership card to the clerk at the counter, anyone can reserve them if such materials are not available in the library at the time. It is possible to have more than one reservation for a single lending item.

- (a) Identify the Actors and the Use Cases one should identify for this system.

(30% Marks)

<b>ANSWER IN THIS BOX</b>	
Actor :	Use Cases
Clerk, Cashier	Selling, Borrowing, Returning,
	Reserving, Preparing Invoice,
	Making Payment





- (b) Identify the potential Classes for the above system.

(30 Marks)

<b>ANSWER IN THIS BOX</b>	
Member	SellingItem
BorrowedCopy	Invoice
ReservedCopy	Payment
Item	Customer
LibraryItem	

- (c) Identify the static relationships between the above classes identified in (b). Indicate the cardinality of the appropriate relationships.

(25% Marks)

<b>ANSWER IN THIS BOX</b>		
Class name1	Class name2	Relationship and UML notation
ReservedCopy	Member	Association 0..* 1..*
LibraryItem	BorrowedCopy	Inheritance 
Item	LibraryItem	Composition 1 0..* 
SellingItem	LibraryMember	Association 0..* 0..1
SellingItem	Invoice	Aggregation 
Invoice	Payment	Association 1 1
Item	SellingItem	Composition 1 0..* 
Member	Invoice	Association 1 1

(d) Identify the object(s) responsible for the Use Case *Reservation* in (a).

(15% Marks)

<b>ANSWER IN THIS BOX</b>
ReservedCopy
Member
LibraryItem

- 2) (a) Given below are statements associated with Object Oriented concepts and UML. You are required to match the statements in **Column X** with those in **Column Y**.

Column X	Column Y
(A) Abstraction	(i) is a relationship where a component can be part of only one whole.
(B) Reflexive Association	(ii) is a special type of relationship that exists if a class consists of a number of component classes.
(C) A composite	(iii) is an expression which says that the objects in the same class can be related to one another.
(D) An Aggregation	(iv) is to filter out an object's properties and operations until just those needed are left.
	(v) is the fundamental conceptual connection between classes.

(20% Marks)

<b>ANSWER IN THIS BOX</b>
A, ( iv )
B, ( iii )
C, ( i )
D, ( ii )

(b) What would be the output of the following program?

```
#include <iostream.h>
class vehicle{
private :int num_wheels;
public :
    vehicle(int w) {
        num_wheels=w;
    }
    vehicle(){cout<<"Vehicle ";}
    void showv(){
        cout<<"Wheels: "<<num_wheels<<"\n";
    }
};

class car:public vehicle{
private: int passengers;
public :
    car(){cout <<"Car ";}
    car(int p, int w):vehicle(w){
        passengers=p;
    }

    void show(){showv();
        cout<<"passengers: "<< passengers<<"\n";
    }
};
```

```
class truck : public vehicle{
private :int loadlimit;
public :
    truck(){cout<<"Truck ";}
    truck(int l, int w):vehicle(w){
        loadlimit=l;
    }

    void show(){showv();
        cout<<"load limit: "<< loadlimit<<"\n";
    }
};

void main() {
    car c(5,4);
    truck t(30000,12);
    cout<<"Car Details:\n";
    c.show();
    cout<<"\nTruck Details:\n";
    t.show();
}
```

(30% Marks)

**ANSWER IN THIS BOX**

Car Details:  
Wheels: 4  
passengers: 5

Truck Details:  
Wheels: 12  
load limit: 30000

(c) If default constructors are removed from the class definitions, what would be the output of the program?

(25% Marks)

**ANSWER IN THIS BOX**

No change.  
Car Details:  
Wheels: 4  
passengers: 5

Truck Details:  
Wheels: 12  
load limit: 30000

(d) Indicate whether the following are *true* or *false*.

(i) The following C++ program will give a compilation error.

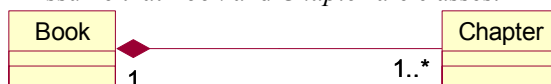
```
#include <iostream.h>
class Rectangle
{
public :
    int m_width,m_height;
    Rectangle(int w, int h)
    {
        m_width=w; m_height=h;
    }
    void DisplayArea(void)
    {
        cout<<m_width*m_height;
    }
    ~Rectangle() { }
};

class NewRectangle : public Rectangle
{
public :
    NewRectangle(int w, int h): Rectangle(w,h){}
    NewRectangle::~~NewRectangle(){}
    DisplayArea()
    {
        cout<<"Area="; cout<<m_width*m_height;
    }
    SetWidth(int w)
    {
        m_width=w;
    }

    SetHeight(int h)
    {
        m_height=h;
    }
};

void main()
{
    NewRectangle MyRectangle(100,20);
    MyRectangle.DisplayArea();
}
```

(ii) The following diagram represents the composition relationship in UML.  
Assume that *Book* and *Chapter* are classes.



(iii) Classes and Use cases are both structural elements and both can inherit.

(iv) The following C++ program will display  $x=1$  as output.

```
#include <iostream.h>
void main()
{
    int x;
    double y=1.7;
    x=y;
    cout<<"x="<<x;
}
```

(v) A use case is the same as a scenario.

(25% Marks)

### ANSWER IN THIS BOX

(i) F	(ii) T	(iii) T
(iv) T	(v) F	

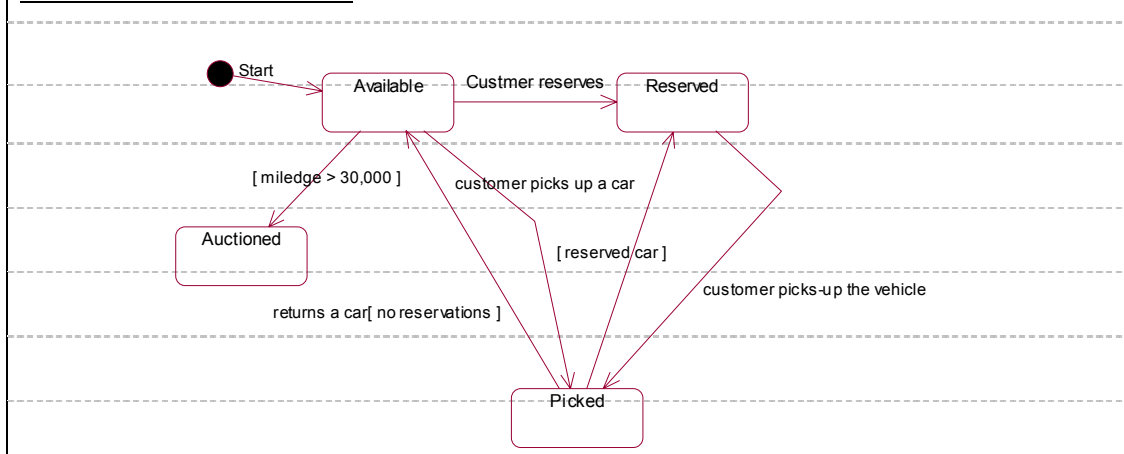
3) The following describes information about a car rental company.

A car Rental company wants to develop an automated system that would handle car reservations, customer billing and car auctions. Usually a customer reserves a car, picks it up and then returns it after a certain period of time. When the car is returned, the customer receives a bill and pays the specified amount. In addition to renting out cars, every year, the company auctions the cars which have accumulated over 30,000 kilometres. For a particular car, there can be many reservations for different time periods and the company may make reservations for a car for future time periods even while it is picked. There can be situations where a car may not have any reservations at all and hence, the car would be truly available for reservations or auctioning.

(a) Draw a state diagram showing all the states and transitions of a car object to include the events which cause the transitions.

(40% Marks)

### ANSWER IN THIS BOX



Following describes the “Car Reservation” scenario of the car rental company described earlier.

When this scenario takes place, assume that a car that satisfies the customers’ expectations and which would be available for the required period is identified by the reservations clerk by a previous use case. No persistence storage is considered in this sequence diagram and assume that all objects are available in memory. A class called ‘ReservationsManager’ is there to control the reservations use case.

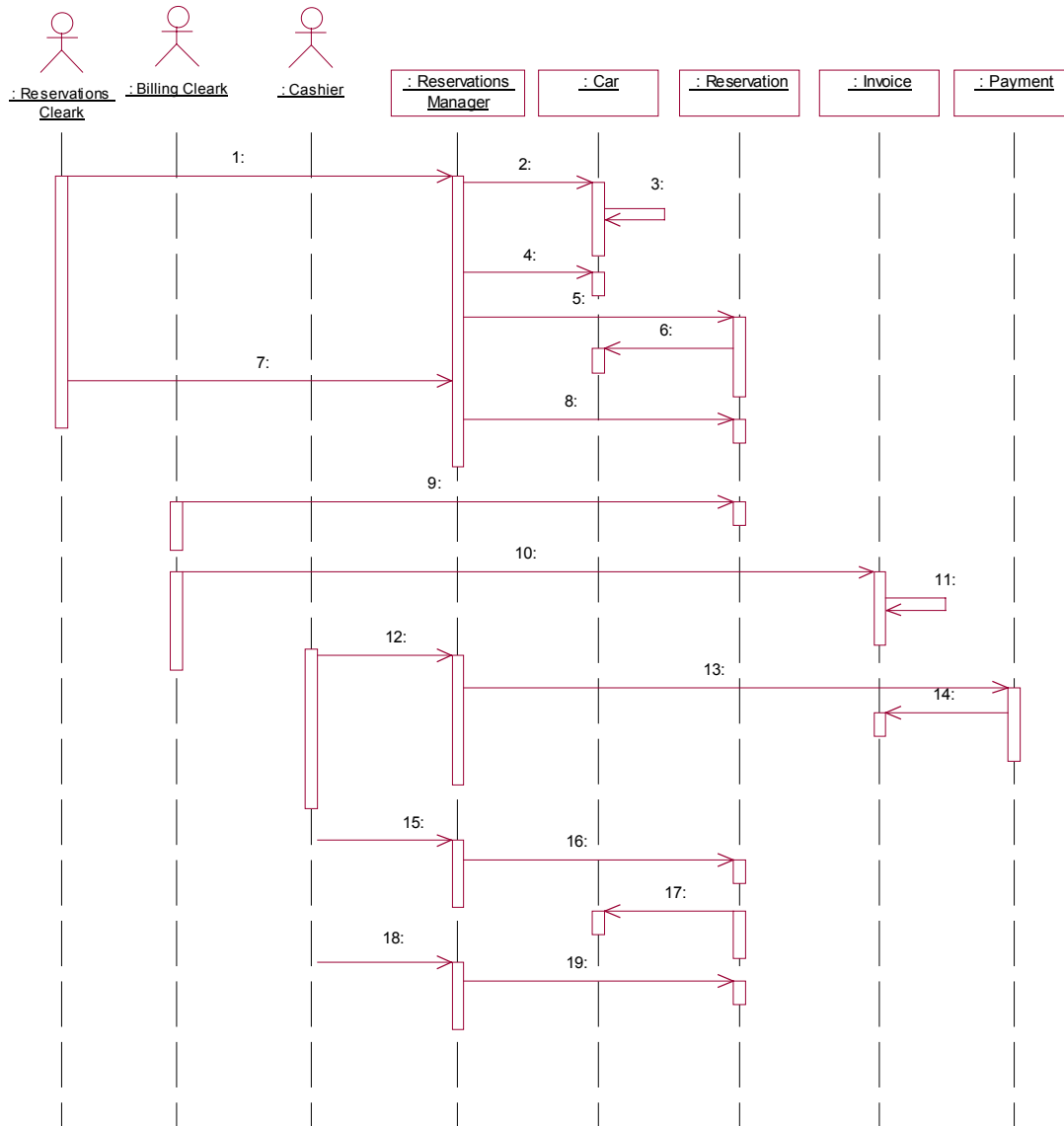
Reservations clerk initiates this scenario by entering the vehicleId, ‘from date’ and ‘to date’ and invoking a method in ‘ReservationsManager’. The appropriate car object is found by the system and a validation is done to ensure whether the car object is available for that period. If that car is available, a preliminary reservation is created and this is also conveyed to the car object. Then the reservations clerk releases this preliminary reservation which can be picked up by the billing clerk.

The billing clerk obtains the released reservation and an invoice object is created for the advance payment due. This invoice is then printed and would be in the printed state.

The customer then produces this invoice at the cashier and makes the payment. The cashier initiates the payment and a new payment object would be created by the system. The invoice would be moved to the ‘Paid’ state. The cashier would then confirm the reservation and the reservations object would be moved to the ‘Confirmed State’ and the car object is also informed about this change. Finally the cashier will print the reservation slip.

(b) The sequence diagram drawn for the reservation scenario is shown below.





Select the appropriate messages from the list given below and fill in the blanks indicated by numbers from 1 to 19.

Messages:

- (a) checkAvailability( )
- (b) releaseReservation( )
- (c) getReleasedReservation( )
- (d) priliminaryReserve( )
- (e) findCar( )
- (f) carDeatils( )
- (g) makePayment( )
- (h) confirmReservation( )
- (i) print( )
- (j) paidState( )
- (k) printReservation( )
- (l) createInvoice( )

- (m) getCarObject( )
- (n) createPriliminaryReservation( )
- (o) release( )
- (p) createPayment( )
- (q) confirm( )

(b) Identify the object(s) responsible for car reservation use case.

(40% Marks)

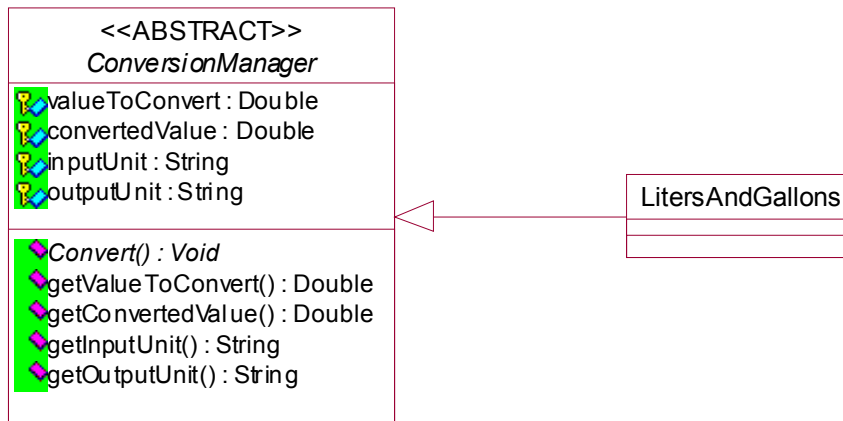
<b>ANSWER IN THIS BOX</b>			
1.	<b>f</b>	10	<b>l</b>
2.	<b>m</b>	11	<b>i</b>
3	<b>e</b>	12	<b>g</b>
4	<b>a</b>	13	<b>p</b>
5	<b>n</b>	14	<b>j</b>
6	<b>d</b>	15	<b>h</b>
7	<b>b</b>	16	<b>q</b>
8	<b>o</b>	17	<b>h</b>
9	<b>c</b>	18	<b>k</b>
		19.	<b>i</b>

(c) List the swim lanes in an activity diagram drawn for the reservation scenario?

(20% Marks)

<b>ANSWER IN THIS BOX</b>
<b>Reservation Clerk</b>
<b>Billing Clerk</b>
<b>Cashier</b>

- 4) A student was assigned the task of developing a simple unit conversion program using C++. He created an abstract class called "ConversionManager" as shown in the following UML diagram.



It contains the protected data members namely ValueToConvert, ConvertedValue, InputUnit, and OutputUnit. Also a constructor which assigns values to each data member, get methods (getValueToConvert, getConvertedValue, getInputUnit, getOutputUnit) to obtain values and a public virtual method called "Convert" which should actually contain the conversion calculation depending on the type of calculation that would take place in the derived classes. For example, if conversion is between Liters and Gallons, then there would be a derived class (LitersAndGallons) as shown in the diagram in which the Convert Method would be overridden. Please note that the datatypes and names of attributes and names, return types, and parameters of operations should be as shown in the diagram.

- (a) What is the condition that should be satisfied for a class to be an abstract class in C++?

(10% Marks)

**ANSWER IN THIS BOX**

To have at least one pure virtual function.

- (b) Define the abstract class 'ConversionManager' in C++. (Implementation of member functions are not required.)

(30% Marks)

**ANSWER IN THIS BOX**

```

class ConversionManager
{
protected:
    double valueToConvert;
    double convertedValue;
    string inputUnit;
    string outputUnit;
public:
    ConversionManager(double valToConv,
                      string inUnit,
                      string outUnit);

    double getValueToConvert();
    double getConvertedValue();
    string getInputUnit();
    string getOutputUnit();
    virtual void Convert()=0;
};
  
```


- (c) Write the implementation of the constructor of the class ‘ConversionManager’ in C++. This should assign 0 to the attribute convertedValue and given values to all other attributes.

(20% Marks)

**ANSWER IN THIS BOX**

```

ConversionManager::ConversionManager(double valToConv,
                                     string inUnit,
                                     string outUnit)
{
    valueToConvert = valToConv;
    convertedValue = 0;
    inputUnit      = inUnit;
    outputUnit     = outUnit;
}

```

- (d) To convert from Litres to Gallons or vice-versa a derived class called LitresAndGallons would be used. Write the definition of the class “LitresAndGallons” in C ++.  
(Implementation of member functions is not required.)

(20% Marks)

**ANSWER IN THIS BOX**

```

class LitersAndGallons : public ConversionManager
{
public:
    LitersAndGallons(double valToConv,
                     string inUnit,
                     string outUnit):ConversionManager(
                                     valToConv,
                                     inUnit,
                                     outUnit)
    {}
    void Convert();
}

```

};

- (e) Implement the convert method in class LitresAndGallons under the following condition.
- If inputUnit is "Litres" and outputUnit is "Gallons" then the conversion formula is  $\text{convertedValue} = \text{valueToConvert} / 3.7854$ .
  - If inputUnit is "Gallons" and outputUnit is "Litres" then the conversion formula is  $\text{convertedValue} = \text{valueToConvert} * 3.7854$ .
  - Otherwise  $\text{convertedValue} = -1.0$

(20% Marks)

**ANSWER IN THIS BOX**

```
void LitersAndGallons::Convert()
{
    if ((inputUnit=="Litres")&&(outputUnit=="Gallons"))
        convertedValue= valueToConvert/3.7854;
    else if ((inputUnit=="Gallons")&&(outputUnit=="Litres"))
        convertedValue= valueToConvert*3.7854;
    else
        convertedValue = -1.0;
}
```

\*\*\*\*\*