UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)
Academic Year 2006/2007 – 2nd Year Examination – Semester 3

IT3102: Object Oriented Systems Development

PART 2 – Structured Question Paper

17th March, 2007
(ONE AND A HALF HOURS)

To be completed by the candidate

BIT Examination Index No: ____________________

Important Instructions:

- The duration of the paper is **1 ½ (one and a half) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **12 pages**.
- **Answer question 1 and any two 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 (×), (e.g. ☒) the numbers of the questions answered.

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

Based on the following case study answer question 1.

1)

Exclusive Corporation

Case Study

Exclusive Corporation is a company that manufactures household electrical appliances.
Information extracted by Systems Analyst and his team is as given below.
The sales catalogue contains 458 items of which more than half are produced by the company.

The following operations were identified.
Order Processing, Inventory Control, Production Scheduling, Manufacturing, Purchasing and Sales.
Due to lack of funds, systems analyst was instructed only to conduct the analysis and design activities for Order processing and Inventory Control operations.
The operations are described in detail below.

Order processing

Each order is accompanied by a 10% deposit. The order processing clerk is responsible for handling the order through the following procedure.

i. First he determines whether the customer is a new or old customer.
ii. If the customer is a new customer, the address, contact number and 2 credit references are taken. Otherwise (if he is an old customer) the credit references are still taken to ensure that the customer is still financially sound.
iii. Each item number, description and price is verified with the catalogue.
iv. The manufactured and purchased items are grouped separately and sent to the Inventory Control Department.
v. When the order-*can be fulfilled* notice is received from Inventory control, a copy of the notice is sent to the Sales Department.

Inventory Control

Inventory includes all raw materials, purchased items and manufactured components. Every item regardless of whether it is purchased or manufactured by the company itself has a maximum and minimum quantity level. These two quantity levels are adjusted based on the popularity of the item and is determined by the management. Every time the inventory control clerk updates an item after dispatching the consignment of items to the customer, he checks whether the remaining quantity falls below the minimum level.

If it does fall below the minimum, the following action is taken.

- If the item is manufactured by the company itself, it sends a product inventory notice to the Product Scheduling Department.
- If it is a purchased item (including raw materials), it sends a purchase order to the Purchasing Department.

When a consignment of raw materials or purchased items is received, the inventory is updated. If the quantity exceeds the maximum quantity, the following action is taken depending on the type of item.

- If it is a manufactured item, an overstock notice is made to management and product scheduling.
- If it is a purchased item, an overstock notice is sent to purchasing.

When customer order reaches inventory control, the following procedure is carried out.

i. The order is broken down into sub-orders for each item.

ii. The order is classified as order *pending*

iii. Each sub-order is checked against the inventory to determine if it can be fulfilled under the current quantities in stock.

iv. If the sub-order can be fulfilled by the inventory the following procedure is carried out.
   - the inventory is adjusted by the given amount
   - the sub-order is filed together with the original pending order

v. If the sub-order cannot be fulfilled, the following action is taken.
   - If the item is manufactured, the sub-order is filed in the manufacturing suspense file and a produce order is sent to the production scheduling department. After manufacturing the item, Inventory Control Department receives a notice from the Manufacturing Department. Subsequently the Inventory is updated.
   - If the item is a purchased item, the sub-order is filed in the purchasing suspense file and a purchase order is sent to the purchase department. After the item is purchased, Inventory Control Department receives a notice from the Purchasing Department. Subsequently the Inventory is updated.

vi. If all the sub-orders can be fulfilled, the order is classified as order *can be fulfilled,* and a notice is sent to the Order Processing Department indicating that the order can be fulfilled.

(a) Identify the Actors and the corresponding Use Cases for the given system.

(40 Marks)

ANSWER IN THIS BOX

Actor: **Order Processing Clerk,**

Use Cases: **Get Customer Details, Verify Orders, Group Orders**

Actor: **Inventory Control Clerk**

Use Cases : **Update Items, Update items after Production, Update items after Purchasing, Update Items after Sales, Check Quantity Levels, Process Sub-Orders,**

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

(30 Marks)

ANSWER IN THIS BOX

Customer, Order, Sub-Order, Item, Purchased Item, Manufactured Item,

Manufacturing suspense details, Purchasing suspense details,

Produce order, Purchase Order

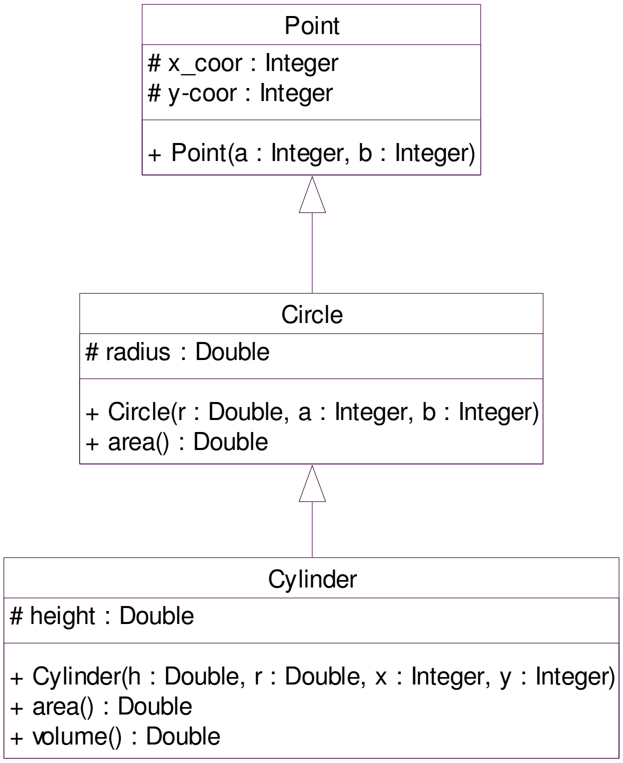
(c) Identify the static relationships between the above classes identified in (b). Indicate the cardinality of the appropriate relationships.

(30 Marks)

ANSWER IN THIS BOX

Class name1	Class name2	Relationship and UML notation
Customer	**Order**	**1- 0..***
Order	**Sub-Order**	**1-1..***
Sub-Order	**Item**	**1- 1**
Item	**Manufactured Item**	**Inheritance**
Item	**Purchased Item**	**Inheritance**
Purchased Item	**Purchasing Suspense file**	**1..0..***
Manufactured Item	**Manufacturing Suspense file**	**1..0..***
Manufacturing Suspense Details	**Produce Order**	**1..1**
Purchasing Suspense Details	**Purchase Order**	**1..1**

2) (a) A student needs to write a program in C++ to draw a Point, Circle and a Cylinder. The class diagram drawn in UML is shown below.



(i) What is the Object Oriented principle used in the above diagram?

(10 Marks)

ANSWER IN THIS BOX

Inheritance

(ii) What are the advantages of using the above concept in systems development?

(15 Marks)

ANSWER IN THIS BOX

code reuse

(iii) Identify all public, private and protected data members and methods.

(15 Marks)

ANSWER IN THIS BOX

Protected - x_coor, y_coor, radius, height

Private - Public - Point, Circle, Cylinder, Area, Volume

(iv) What is the advantage of defining the data member 'x_coor' with the given scope?

(15 Marks)

It can be accessed only by the classes

inherited from Point class.

(b) The Area of a Circle, the Area of a Cylinder and the Volume of the Cylinder can be found with the use of the following formulae:

Area (Circle) = PI * radius * radius
Area (Cylinder) = 2 * Area (Circle) + (2 * PI * radius * height)
Volume (Cylinder) =Area (Circle) * height

It is required to map the above class diagram to C++ code.

(i) Write down the C++ definition of class Point and the constructor implementation

(20 Marks)

ANSWER IN THIS BOX

```
class Point{

                public:           Point(int, int);

                protected:        int x_coor, y_coor;

                };

                Point::Point(int a,int b)

                {  x_coor=a;

                   y_coor=b;

                }
```

(ii) Write the C++ definition of class Circle and the Constructor implementation.

(20 Marks)

ANSWER IN THIS BOX

```
class Circle : public Point{
                    public: Circle(double, int, int);
                            double area();
                    protected: double radius;
          };
          Circle::Circle(double r, int a, int b) :Point(a,b)
          {
              radius=r;
          }
```

(iii) Write the C++ definition of class Cylinder and the constructor implementation.

(20 Marks)

ANSWER IN THIS BOX

```
class Cylinder :public Circle {
public:             Cylinder(double, double, int, int);
                    double area();
                    double volume();
    protected:      double height;
};
Cylinder::Cylinder(double h, double r, int x, int y) : Circle ( r, x, y)
{
    height = h;
 }
```

3) Flow of Events Document for the *Loan* Scenario for a Library System is given below.

Main Flow

```
Librarian enters the borrower id.

System checks whether borrower id exist.
    If id does not exist (E-1) end use case.
Else
    Check for Overdue Books.
    If  there are Overdue books
        Display Overdue details
        (E-2)
     end use case

    Check Over limit (E-3)
    If  reached over limit
        Display Overdue Details
        (E-3)
     end use case

Enter copy id
    Check Copy Exist
     If not (E-4) end use case

Check Borrow able
     If  it is reference (E-5) end use case

Librarian Confirm Borrowing  (C-5)

Update Borrowed Copy details
```

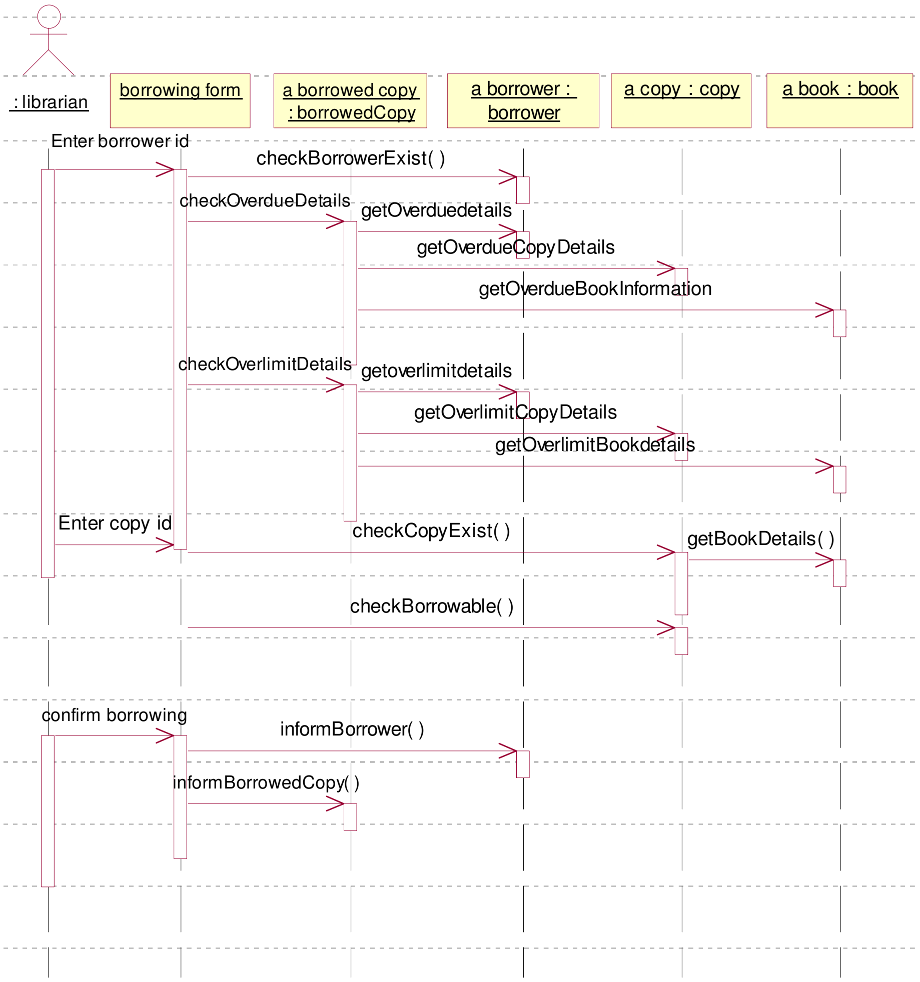
Alternate Flows

E-1 : Borrower id does not exist
E-2 : There are overdue books
E-3 : Borrower has already borrowed 5 books
E-4 : Copy does not exist
E-5 : Copy is not borrowable
C-1 : Confirm borrowing Message box

Draw a Sequence Diagram for the given Flow of Events Document for Loan Scenario. Assume that the objects which need to interact with this scenario are Book (Title, Author, etc.), Copy (CopyId, CopyType, Price, etc.), Borrower (BorrowerId, Name, etc.) and BorrowedCopy(BorrowerId, CopyId, BorrowedDate, ReturnedDate).

(100 Marks)

ANSWER IN THIS BOX



4) A payroll system to perform payroll calculations based on the type of an employee, needs to be created. The class ‘Employee’ should contain two protected attributes, namely *firstname* and *lastname*. The two attributes are of type String. The two argument constructor sets values to the two attributes. Two pure virtual functions are also defined in this class as *earnings* and *print*. Function *earnings* with return type double, will return the earning of the employee. *Print* function will output the employee’s name.

The derived classes of Employee are ‘Boss’ who gets paid a fixed weekly salary regardless of the number of hours worked, and ‘HourlyWorker’ who gets paid by the hour and receives overtime pay.

The class ‘Boss’ contains a private attribute called *weeklySalary* of type double and there is a constructor with three arguments to set the *firstName*, *lastName* and *weeklySalary* in class Boss.

The class ‘HourlyWorker’ contains two private attributes *wage* and *hours* of type double. *Wage* to store the wage per hour and *hours* to store the hours worked per week. A constructor also there with four arguments to set the attributes *firstName*, *lastName*, *wage* and *hours*.

(a) What type of a class is Employee in terms of C++? Is ‘earnings’ a pure virtual function? Give reasons for your answer.

(10 Marks)

ANSWER IN THIS BOX

Abstract class – Because of the pure virtual function

earnings pure virtual – Because we cannot calculate the earnings for a generic employee.

(b) Write down the definition for class Employee in C++. Implementations are not needed.

(20 Marks)

ANSWER IN THIS BOX

```
class Employee{
          public:
               Employee(const char *, const char *);
               virtual double earnings() = 0;
               virtual void print();
          protected:
               char *firstName;
               char *lastName;
   };
```

(c) Define the class Boss in C++. Implementations are not needed.

(30 Marks)

ANSWER IN THIS BOX

```
class Boss : public Employee{
        public:
            Boss(const char *, const char *, double=0.0);
            virtual double earnings();
            virtual void print();
        private:
            double weeklySalary;
};
```

(d) Define the class HourlyWorker in C++. Implementations are not needed.

(30 Marks)

ANSWER IN THIS BOX

```
class HourlyWorker : public Employee{
        public:
        HourlyWorker (const char *, const char *, double=0.0, double=0.0);
        virtual double earnings();
        virtual void print();
            private:
                double wage;
                double hours;
    };
```

(e) Print() function within the derived classes are used to output the type of Employee and then calls print() function within the Employee to output the employee's name . Write the implementation of the overridden function print() in Boss Class.

(10 Marks)

ANSWER IN THIS BOX

```
void Boss::print()

        {

                cout<<"\n  Boss ";

                Employee::print();

        }
```

****