

**UNIVERSITY OF LONDON**

**GOLDSMITHS COLLEGE**

**B.Sc. Examination 2002**

**COMPUTING AND INFORMATION SYSTEMS**

**IS52004A (CIS210) Software Engineering and Development**

**[Internal]**

Duration: 3 hours

- 
- *Full marks will be awarded for complete answers to FOUR questions. Do not attempt more than FOUR questions on this paper.*
  - *Electronic calculators may be used. The make and model should be specified on the script. The calculator must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.*

**THIS EXAMINATION PAPER MUST NOT BE REMOVED  
FROM THE EXAMINATION ROOM**

**Question 1.**

- a) What does the project management spectrum involve? [4]
- b) Name three key participants in the software management process.  
Explain the role of each of them. [6]
- c) What are the six subprocesses in the organisation of the software management process?  
Give a brief explanation of the activities included in each subprocess. [12]
- d) Name the three main project estimates necessary to perform software planning? [3]

**Question 2.**

Perform software project estimation using the FunctionPoints metric assuming the following software information domain characteristics:

<i>Measurement parameter</i>	<i>count</i>	<i>factor</i>
number of user inputs	20	4
number of user outputs	15	5
number of user inquiries	28	5
number of files	5	10
number of external interfaces	3	7

Suppose that the software complexity adjustment values are as follows:

<i>Complexity adjustment</i>	<i>value (F<sub>i</sub>)</i>
Backup and recovery	5
Data communications	4
Distributed processing	2
Performance critical	5
Existing operating environment	3
On-line data entry	3
Input transaction over multiple screens	5
Master files updated on-line	2
Information domain values complex	4
Internal processing complex	5
Code designed for reuse	3
Conversion/installation in design	4
Multiple installations	5
Application designed for change	5

- Calculate the estimated number of function points  $FP_{est}=?$  [9]
- Compute the productivity of the team for 3 person-months using the derived function points  $FP_{est}$ . [4]
- Compute the quality of this software having 5 defects using the derived function points  $FP_{est}$ . [4]
- Compute the software cost using the derived function points  $FP_{est}$  if there have been spend 3000 pounds. [4]
- Compute the documentation size using these function points  $FP_{est}$  if 100 pages have been produced. [4]

**Question 3.**

- a) Concerning the notion of a class in context of object-oriented software engineering, answer the following questions:
- i) What are the main components of a UML class diagram? [2]
  - ii) How do we represent a class in a UML class diagram? [2]
  - iii) What relationships do we specify with: association, aggregation, and inheritance? [6]
- b) Develop a class diagram in terms of the UML language depicting class characteristics and relationships according to the following description of *Fred's Dodge dealership sales system*:

The vehicle hierarchy in the sales system the abstract class `Vehicle` has two subclasses: `FamilyCar` and `RecreationalVehicle`; the class `Mini-Van` inherits from `RecreationalVehicle`.

A private attribute of the `Vehicle` class is `stock`; a public method is `ReturnCost`. The `Vehicle` class serves to keep track of the basic public information about the `enginepower`, `maxspeed`, and `fuelconsumption`, which next may be provided for storing into the database. The available vehicles are maintained in a public class variable `listofvehicles` in a `VehicleDatabase`. The particular `vehicleitems` could be loaded with information about their engine power, maximal speed, and fuel consumption.

The `InventoryController` has the responsibility of tracking sales of each vehicle and maintaining an appropriate supply. Aggregated within `InventoryController` is one supporting class `MarketAnalysis`. The `MarketAnalysis` has private functions for `CalculateDaylySales` and `CalculateMonthlySales`, as well as a public function `ReturnSalesInfo`. [15]

**Question 4.**

- a) i) Define the notion of cyclomatic complexity of a software program? [2]  
ii) Is there any relation between the cyclomatic complexity of a software program and the number of independent paths in the program? [2]  
iii) How do we use the cyclomatic complexity measure in basis path testing? [3]
- b) Determine the basis set of linearly independent testing paths in the program fragment given below as follows:  
i) draw the flow graph of this procedure;  
ii) estimate the cyclomatic complexity of the flow graph;  
iii) find the basis set of linearly independent testing paths. [18]

```
int binary( struct node a[],int N, int v )
{
    int l; int r; int x; int res = -1;

    l=1; r=N;
    while ( r >= l )
    {
        x = ( l + r ) / 2;
        if ( v < a[ x ].key )
            r = x-1;
        else
            l = x+1;
        if ( v == a[ x ].key )
            res = a[ x ].info;
    }
    return res;
}
```

**Question 5. Activity Diagrams**

- a) What are the functions of activity diagrams in System Development? [4]
- b) Draw an activity diagram for the task of making a pot of tea, and pouring out a cup with or without milk and with or without sugar. [13]
- c) What are swim lanes and how can they be used to make activity diagrams more expressive? [4]
- d) Redraw your activity diagram assuming that you have two actors: "Mother" who is in charge of the teapot and "Father" who is in charge of the cups. [4]

**Question 6. Use Cases**

a) What are the functions of use case diagrams in system development. **[5]**

b) Draw a high-level use case diagram for a system meeting the following requirements:

The system will be a web-based toy shop. It can be accessed by anybody with internet access. A customer can pay for a purchase by giving a credit card number; an account holder can pay for a purchase by drawing the money from his account. An account holder may at anytime deposit money into his account. Any customer can open an account using a credit card with a minimum deposit of 25 pounds. The toy shop accounting system checks credit cards and keeps tracks of accounts. **[10]**

c) Explain the use of the stereotypes in use case diagrams. **[2]**

d) Redraw your diagram including an instance of each type of stereotypes. Explain why the stereotypes are appropriate in your example. **[8]**

**UNIVERSITY OF LONDON**

**GOLDSMITHS COLLEGE**

**B.Sc. Examination 2002**

**COMPUTING AND INFORMATION SYSTEMS**

**IS52004A (CIS210) Software Engineering and Development**

**[Solutions Internal]**

Duration: 3 hours

- 
- *Full marks will be awarded for complete answers to FOUR questions. Do not attempt more than FOUR questions on this paper.*
  - *Electronic calculators may be used. The make and model should be specified on the script. The calculator must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.*

**THIS EXAMINATION PAPER MUST NOT BE REMOVED  
FROM THE EXAMINATION ROOM**



# Solutions CIS 210 Internal

## Question 1.

a) What does the project management spectrum involve? [4]

Project management involves the planning, monitoring and control of the people, process, and events that occur as software evolves from a concept to implementation.

b) Name three key participants in the software management process.  
Explain the role of each of them. [6]

The software process is populated by the following players:  
senior managers, project managers, practitioners, and customers.

The senior manager coordinates the connection between the business and the software professionals.  
The project manager plans, monitors, and controls the work of a team of software engineers.  
The software practitioner (engineer) manages his day to day activities, planning, and managing technical tasks.

c) What are the six subprocesses in the organisation of the software management process?  
Give a brief explanation of the activities included in each subprocess. [12]

The subprocesses in the organisation of the software management process are:

1. Beginning a Software Project- includes establishment of the software objectives and goals;
2. Identification of measures- includes selection of proper project measures and metrics for getting insights into the organisation of the software engineering;
- 3 Project Estimation- usually requires to broke the project into pieces;
- 4 Risk Analysis- includes risk identification, assessment, prioritisation, management, and monitoring;
- 5 Project Scheduling- includes identification of the project tasks, and development of a time-schedule;
- 6 Tracking and Control- If the task falls behind the schedule, the manager tries to determine the impact of project slippage on the project milestones, and the overall delivery rate.

d) Name the three main project estimates necessary to perform software planning? [3]

The three main project estimates necessary to perform planning are:  
human effort, project duration, and cost.

**Question 2.**

Perform software project estimation using the FunctionPoints metric assuming the following software information domain characteristics:

<i>Measurement parameter</i>	<i>count</i>	<i>factor</i>
number of user inputs	20	4
number of user outputs	15	5
number of user inquiries	28	5
number of files	5	10
number of external interfaces	3	7

Suppose that the software complexity adjustment factors are as follows:

<i>Complexity adjustment</i>	<i>value (F<sub>i</sub>)</i>
Backup and recovery	5
Data communications	4
Distributed processing	2
Performance critical	5
Existing operating environment	3
On-line data entry	3
Input transaction over multiple screens	5
Master files updated on-line	2
Information domain values complex	4
Internal processing complex	5
Code designed for reuse	3
Conversion/installation in design	4
Multiple installations	5
Application designed for change	5

a) Calculate the estimated number of function points  $FP_{est}$ ? [9]

$$FP_{est} = count\text{-total} \times [ 0.65 + 0.01 \times SUM(F_i) ]$$

$$FP_{est} = 366 \times [ 0.65 + 0.01 \times 55 ] = 439.2$$

b) Compute the productivity of the team for 3 person-months using the derived function points  $FP_{est}$ . [4]

$$Productivity = FP_{est} / person\text{-month}$$

$$Productivity = 439.2 / 3 = 146.4 \text{ [function points per person month]}$$

c) Compute the quality of this software having 5 defects using the derived function points  $FP_{est}$ . [4]

$$Quality = defects / FP_{est}$$

$$Quality = 5 / 439.2 = 0.011 \text{ [defects per function point]}$$

d) Compute the software cost using the derived function points  $FP_{est}$  if there have been spend 3000 pounds. [4]

$$Cost = \$ / FP_{est}$$

$$Cost = 3000 / 439.2 = 6.83 \text{ [pounds per function point]}$$

e) Compute the documentation size using these function points  $FP_{est}$  if 100 pages have been produced. [4]

$$Documentation = pages \text{ of documentation} / FP_{est}$$

$$Documentation = 100 / 439.2 = 0.228 \text{ [pages per function point]}$$

**Question 3.**

a) Concerning the notion of a class in context of object-oriented software engineering, answer the following questions:

i) What are the main components of a UML class diagram? [2]

The UML class diagram shows classes together with the possible relationships between them.

ii) How do we represent a class in a UML class diagram? [2]

A class in UML is drawn with a rectangle, divided into three compartments: name compartment, attribute compartment, and operations compartment.

iii) What relationships do we specify with: association, aggregation, and inheritance? [6]

Association denotes some semantic dependence among classes.

Aggregation indicates that the relationships between the classes is a sort of "contains", "part-of".

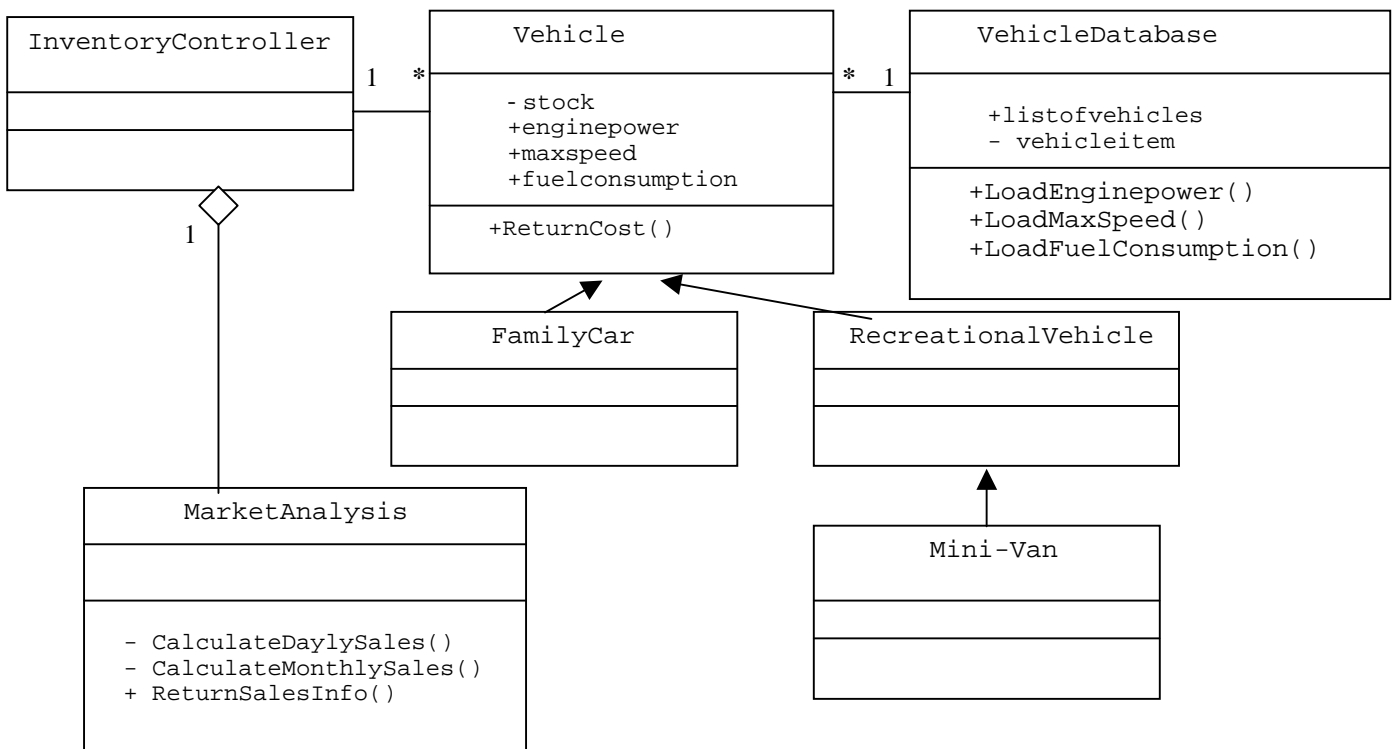
Inheritance denotes vertical "is-a" relationships between classes.

b) Develop a class diagram in terms of the UML language depicting class characteristics and relationships according to the following description of *Fred's Dodge dealership sales system*:

The vehicle hierarchy in the sales system the abstract class `Vehicle` has two subclasses: `FamilyCar` and `RecreationalVehicle`; the class `Mini-Van` inherits from `RecreationalVehicle`.

A private attribute of the `Vehicle` class is `stock`; a public method is `ReturnCost`. The `Vehicle` class serves to keep track of the basic public information about the `enginepower`, `maxspeed`, and `fuelconsumption`, which next may be provided for storing into the database. The available vehicles are maintained in a public class variable `listofvehicles` in a `VehicleDatabase`. The particular `vehicleitems` could be loaded with information about their engine power, maximal speed, and fuel consumption.

The `InventoryController` has the responsibility of tracking sales of each vehicle and maintaining an appropriate supply. Aggregated within `InventoryController` is one supporting class `MarketAnalysis`. The `MarketAnalysis` has private functions for `CalculateDaylySales` and `CalculateMonthlySales`, as well as a public function `ReturnSalesInfo`. [15]



#### Question 4.

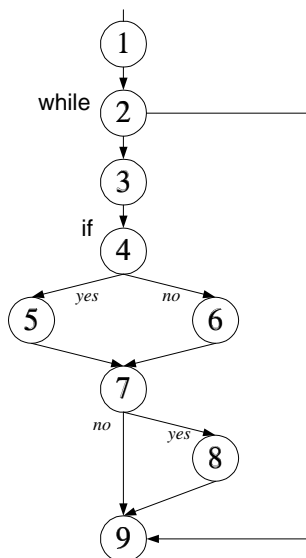
- a) i) Define the notion of cyclomatic complexity of a software program? [2]
- ii) Is there any relation between the cyclomatic complexity of a software program and the number of independent paths in the program? [2]
- iii) How do we use the cyclomatic complexity measure in basis path testing? [3]

The cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program.

An independent path is any path through the program that introduces at least one new statement or a new condition. In terms of a flow graph, an independent path must move along at least one edge that has not been traversed before. The number of independent paths in a program is measured by the cyclomatic complexity of that program.

The value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides the upper bound for the number of tests that must be conducted in order to guarantee that all statements have been executed at least once.

- b) Determine the basis set of linearly independent testing paths in the program fragment given below as follows:
  - i) draw the flow graph of this procedure;
  - ii) estimate the cyclomatic complexity of the flow graph;
  - iii) find the basis set of linearly independent testing paths. [18]



The basis set contains 4 of linearly independent testing paths, since the cyclomatic complexity of this program fragment is four (as there are obviously four regions in the above flow graph):

Path 1: 1-2-9

Path 2: 1-2-3-4-5-7-9

Path 3: 1-2-3-4-6-7-9

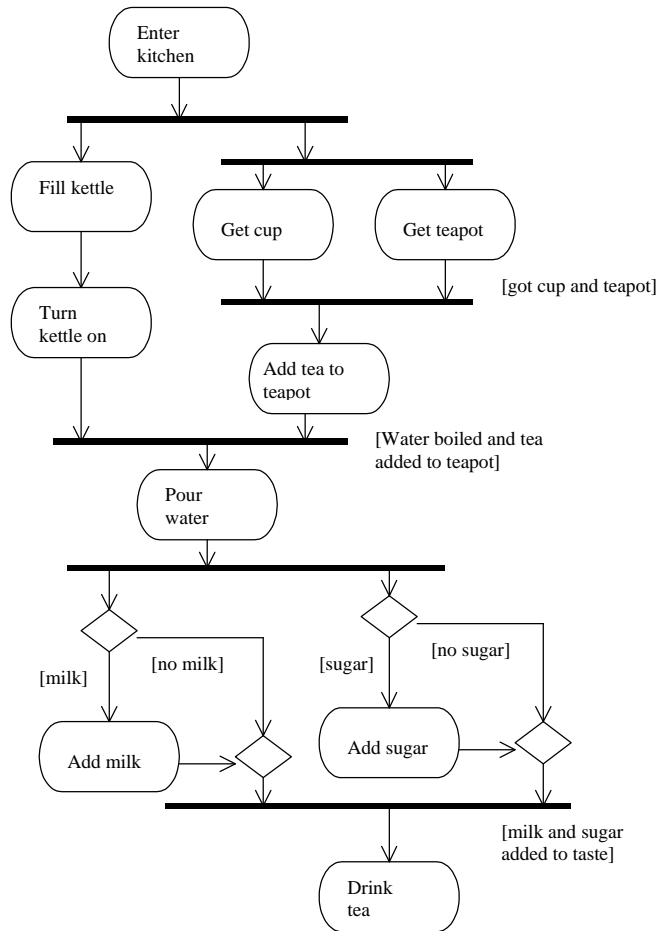
Path 4: 1-2-3-4-5-7-8-9

**Question 5. Activity Diagrams**

a. What are the functions of activity diagrams in System Development? [4]

We can use an **activity diagram** to model the co-ordination and sequencing of actions in order to achieve a given purpose and, with a simple extension, show the changes in responsibility along the way. Use them to help you investigate the flow of control from one activity to another. Activity diagrams are one of the five diagrams in the UML that allow you to model dynamic aspects of a software system.

b. Draw an activity diagram for the task of making a pot of tea, and pouring out a cup with or without milk and with or without sugar. [13]



c. What are swim lanes and how can they be used to make activity diagrams more expressive? [4]

The set of activities in a given diagram may contain several groups of related activities. For example, they may be associated with the actors that perform them. Each partition or swimlane shows which actor is responsible for which set of related activities. The same idea can be used for assigning responsibilities to objects, although an interaction diagram is likely to be more useful at the implementation level.

d. How can you use swimlanes to make your tea making activity diagram more expressive. [4]

You could say that there are two people responsible for parts of the tea making. Mother, who is the crockery person, and Father who is the Kettle person. Swimlanes could then be used to demarcate their individual responsibilities. It is a bit absurd in this example, but it does make it clear what can be done in parallel by different people.

**Question 6.** Use Cases

a. What are the functions of use case diagrams in system development.

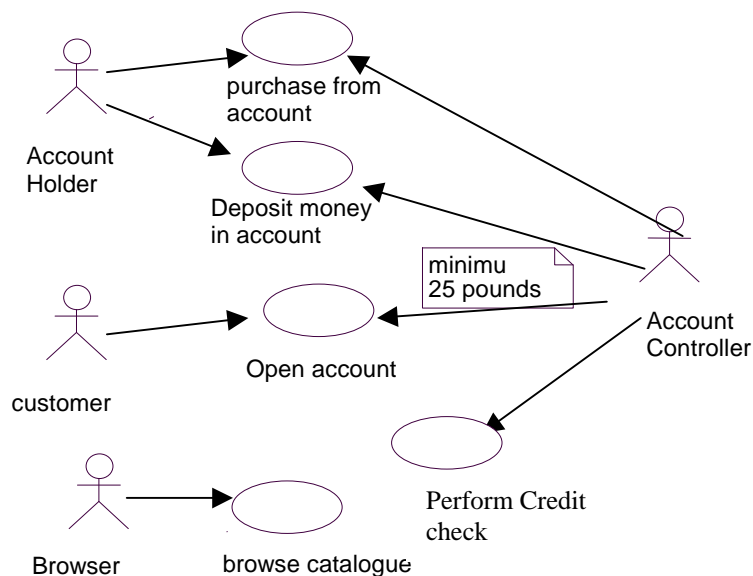
A high-level use case model describes the required (software) system from the users' point of view, though developers are more likely to produce them. [3]

A slightly lower-level one may include initial design decisions in the form of having some use cases include others, or be modelled as variants of others. [2]

b. Draw a high-level use case diagram (no use of stereotypes) for a system meeting the following requirements:

The system is for a web-based toy shop. The catalogue can be browsed by anybody with internet access. To make a purchase you must have an account. You then purchase a toy by drawing the money from his account. An account holder may at any time deposit money into his account. Any customer can open an account using a credit card.. The toy shop accounting system checks credit cards and keeps tracks of accounts.

Note: you do not need to model the workings of the toy shop—e.g. updating catalogues, sending off toys etc.—but only the outside users and the account controller. [10]



*5 marks for the general idea (use cases are verb phrases, right shapes etc.);  
5 marks for getting it nearly right*

c. What are the two stereotypes commonly used in use case diagrams. [2]

The two different stereotypes used are <<include>> and <<extend>> / 1 mark each /

d. The toy shop is having a special promotion: every thousandth customer gets a free toy. Redraw enough of your diagram to include the promotion and an instance of each type of stereotype. [8]

