W'08 : 2 FN : CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** *and* ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answer may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification.

Figures on the right-hand side margin indicate full marks.

Group A

(*a*) What do you understand by a software life cycle model? What problems might occur if a software development organization does not use any specific life cycle model? 6

(*b*) What is the difference between requirements analysis and requirements specification? Using suitable examples, explain different types of requirement problems that should be identified and resolved during the requirements analysis activity. 8

(*c*) Is it true that a software product can always be developed faster by having a larger development team of competent software engineers? Justify your answer

2. (*a*) Besides code commenting, explain different ways in which a program source code can be documented. 7

(*b*) What are the advantages and disadvantages of formal specification over traditional specification? Give at least one example for which formal specification is desirable and another for which formal specification is undesirable. 7

(*c*) What do you understand by the term 'top-down decomposition' in the context of function-oriented design? 6

3. (*a*) What do you mean by the term 'data dictionary' in the context of structured analysis? How is the data dictionary useful in different phases of the life cycle of a software product? 6

(*b*) Draw a labelled DFD model (context diagram, level 1 diagrams, and data dictionary only) for a student academic management software that should support the following features: 14

- Register students to subjects
- Award marks for subjects
- Query marks
- Print report cards
- Compute statistics of student performance.

4. (*a*) What do you understand by the term 'phase containment of errors'. Why is it important to have phase containment of errors? 6

(*b*) What do you understand by information hiding? Explain why a design approach based on the information hiding principle is likely to lead to a reusable and maintainable design. Illustrate your answer with a suitable example. 7

(*c*) What do you mean by cohesion and coupling in a design? Is it true that a good design should have high coupling and low cohesion? Explain your answer. 7

Group B

5. (*a*) What is the difference between a typeless, weakly typed, and strongly typed programming languages? What are the relative advantages of each? Give examples of each type of language? 10

(*b*) Explain important concepts in object-orientation using suitable examples. What are the reasons behind increased productivity when the object-oriented paradigm is adopted? 10

6. (*a*) What is the difference between black-box and white-box testing? Can one be used in place of another? 7

(*b*) Explain why measurement of software reliability is a much harder problem than the measurement of hardware reliability. 6

(*c*) Distinguish between software verification and software validation. When are verification and validation performed during the software life cycle? Can one be used in place of the other? 7

7. (*a*) List important shortcomings of LOC for use as a software size metric. Does the function point metric overcome these? Explain your answer. 8

(*b*) List four metrics that can be determined from an analysis of a program's source code and would correlate well with the reliability of the delivered softwaré. 4

(*c*) What do you understand by software quality assurance? How is it achieved? 8

8. Consider the following program segment:

```
int find-maximum (int i, int j, int k) {
 int max;
   if (i>j) then
    if (i>k) then max=i;
            else max=k;
      else if (j>k) max=j
            else max=k;
    return (max);
   }
```

(*a*) Draw the control flow graph for this program segment. 6

(*b*) Determine the cyclomatic complexity for this program. Show the intermediate steps in your computation. Writing only the final result is not sufficient. 7

(*c*) How is the cyclomatic complexity metric useful? 7

Group C

9. Answer whether the following statements are *true* or *false*. In each case, justify your choice in one sentence: 2 × 10

(*i*) Error and failure are synonymous in software testing terminology.

(*ii*) Size of a project, as used in COCOMO, is the size of the final executable code in bytes.

(*iii*) According to the COCOMO model, cost is the most fundamental attribute of a software product, based on which size and effort are estimated.

(*iv*) The number of development personnel required for any software development project can be obtained by dividing the total (estimated) effort by the total (estimated) duration of the project.

(*v*) Modern quality assurance paradigms are centered around carrying out thorough product testing.

(*vi*) A pure top-down integration testing does not require the use of any stub modules.

(*vii*) During detailed design, the algorithms and data structures are designed.

(*viii*) The extent to which two modules depend on each other determines the cohesion of two modules.

(*ix*) A structure chart is essentially the same as a flow chart.

(*x*) Halstead's effort equation can be easily applied during the planning stage of a software development project, since the equation can be easily automated.

S'09 : 2 FN : CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** *and* ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answer may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification.

Figures on the right-hand side margin indicate full marks.

Group A

1. (*a*) Explain the following terms with respect to software design: (*i*) Data design, (*ii*) architectural design, and (*iii*) detailed design. 3 × 2

(*b*) Explain what is meant by a decision table. How is it useful? Also, give one example of decision table. 6

(*c*) Develop a procedural design for a program that reads text file and produces following output: 4

(*i*) No. of words in the file, and

(*ii*) No. of characters in the file.

(*Turn Over*)

(*d*) What is the difference between a flow chart and a data flow diagram? 4

2. (*a*) Define the following terms with respect to data flow-oriented design: (*i*) Transform flow, and (*ii*) transaction flow. 8

(*b*) For a computer-based library information system, draw the first two levels of data flow diagram. 12

3. (*a*) How is the quality of a software judged? How is software quality achieved in practice? 6

(*b*) Define the following terms in brief: (*i*) β testing, and (*ii*) integration testing. 2+2

(*c*) Explain the iterative waterfall model of software development with the help of a diagram. 10

4. (*a*) What is meant by SRS document? What are the characteristics of a good SRS document? 6

(*b*) Draw a level O DFD of invoice creation system. Assume the component as dealer, customer, head-office, product file, dealer file, customer file, order file, invoice file, stock book. You may assume any other data, if required. 6

(*c*) Define the following: 4 × 2

(*i*) Data coupling

(*ii*) Control coupling

(*iii*) Common coupling

(*iv*) Content coupling.

Group B

5. (*a*) Describe the following debugging approaches in brief: 10

(*i*) Brute force technique

(*ii*) Back tracking approach

(*iii*) Cause elimination approach.

(*b*) Describe top down and bottom up approaches of integration testing. Give an example of each. What are the roles of stubs and drivers in integration testing? 10

6. (*a*) What are the applications of Gantt chart and PERT diagram? Explain the important features of these diagrams using suitable examples. 10

(*b*) What is meant by software maintenance? What are the various types of maintenance? Explain them. 10

7. Define the following in brief in context of software quality: (*i*) Integrity, (*ii*) usability, (*iii*) maintainability, (*iv*) testability, (*v*) portability, (*vi*) reusability, (*vii*) interoperability, (*viii*) consistency, (*ix*) execution efficiency, (*x*) expandability. 10 × 2

8. Write short notes on *any four* of the following: 4 × 5

(*i*) Code review techniques

(*ii*) Software quality management system

(*iii*) SEI capability maturity model

(*iv*) CASE tools

(*v*) Software documentation.

Group C

9. Fill in the blanks :

(*i*) A cohesive module performs ——.

(*ii*) Objective C and small talk are examples o languages.

(*iii*) DFD stands for ——.

(*iv*) Normalization in relational database des done to ——.

(*v*) An assembler converts —— into ——.

(*vi*) Data design deals with ——.

(*vii*) Architectural design deals with ——.

(*viii*) —— checks the consistency of data.

(*ix*) —— tools are used for designing the sc modules.

(*x*) —— indicates how closely two modules i between themselves.

W'11:2 FN:CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** and ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answers may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification

Figures on the right-hand side margin indicate full marks.

Group A

1. (*a*) Explain the prototyping software development life-cycle model using a suitable schematic diagram. 10

(*b*) How is a software project planned ? Briefly explain. 10

2. (*a*) Compare top-down and bottom-up approaches to design. 10

(*b*) Explain the sequence of events that take place when you use your ATM card to withdraw money. 10

3. (*a*) Represent the following using a decision table. There are three types of tax-payers—male, female and senior citizen. A male tax-payer pays 10% tax for income of more than Rs 1,60,000 and less than Rs. 3,00,000. A female pays 10% tax, if her income is more than Rs. 1,80,000 and less than Rs. 3,00,000, and a senior citizen pays 10% tax, if his income is more than Rs. 2,10,000 and less than 3,00,000. If a tax-payer income is more than 3,00,000 and less than 6,00,000, he has to pay 10% tax up to 3,00,000 + 20% tax on the income which is more than 3,00,000. If the income is more than 5,00,000 and less than Rs.10,00,000, then he has to pay tax up to income Rs. 5,00,000 + 30% tax on the income which is more than Rs. 5,00,000. If the income is more than Rs. 10,00,000, then a tax-payer has to pay tax on the income up to Rs. 10,00,000 + 40% tax on the income higher than 10,00,000. 10

(*b*) Explain cohesion and coupling of programs with examples. How does this concept help design good programs ? 10

4. (*a*) List advantages of using structured coding. What are the advantages of structured programming ? 10

(*b*) Explain the following terms : 2 + 2 + 2 + 4

(*i*) Software integration

(*ii*) Software documentation

(*iii*) Performance testing

(*iv*) Reliability growth modelling.

Group B

5. (*a*) What is the difference between program verification and validation. 5

(*b*) State the advantages of using high level languages as compared to a machine language or assembly language. 5

(*c*) What is the difference between a strongly typed, weakly typed and typedon language ? What are their relative advantages ? Give examples of each. 10

6. (*a*) State and explain concepts of object-oriented design. What is an object based programming language ? 10

(*b*) What do you understand by 'concurrency' and 'multithreading' ? Give an example of concurrency in an application. 10

7. (*a*) Compare white box testing and black box testing. 10

(*b*) What is a code metric ? Explain different code metrics. 5

(*c*) Design black box test cases for a function that searches for a given number in an array of 100 numbers. 5

8. (*a*) Explain different software quality parameters. 5

(*b*) Explain how the cost of a software is estimated. Give an example of estimation. 10

(*c*) Write *five* salient requirements for obtaining ISO 9000 certification. 5

Group C

9. Answer the following in brief : 10 × 2

(*i*) What is meant by a functional requirement ?

(*ii*) Define recursion.

(*iii*) What do you understand by the reliability of a software ?

(*iv*) What do you mean by data encapsulation ?

(*v*) Define cyclomatic metric.

(*vi*) What is a module of a program ?

(*vii*) What is a pseudo code ?

(*viii*) What is software engineering ?

(*ix*) What is re-engineering ?

(*x*) What do you understand by a metric ?

S'12: 2FN: CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** and ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answers may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification

Figures on the right-hand side margin indicate full marks.

Group A

1. (*a*) What is planned during software project planning ? Enumerate the steps through which planning is achieved. 7

(*b*) How is software different from a program ? Write a brief note on software crisis. 6

(*c*) Explain the phases in an iterative software development process. 7

2. (*a*) What do you mean by a good SRS ? Discuss the characteristics of a good SRS. 8

(*Turn Over*)

(*b*) A library database contains entries that have the name of the book, followed by author's name, the publisher's name, the year of publication of the book, and the ISBN no. of the book. It also contains the number of copies of the book. Each of the data entries is on a new line. Represent this database as a regular expression. 4

(*c*) Explain how Petri nets can be used to design concurrent processing. 8

3. (*a*) What is meant by modular design ? How can the modularity of a design be assessed ? 8

(*b*) Discuss the role of cohesion and coupling in software design. 5

(*c*) Distinguish between DFDs and structure charts. Explain, with a suitable illustrative example, how a DFD can be converted to a structure chart. 2 + 5

4. (*a*) What is meant by structured programming ? Give an example of an 'unstructured' program. What are the advantages of structured programming ? 2 + 4 + 4

(*b*) Why is documentation important ? What are different documentation that must accompany a software product ? What are the different ways in which a program code can be documented ? 10

Group B

5. (*a*) What is the difference between strongly typed, weakly typed, and typeless languages ? Give one example of each of these categories of programming languages. 10

(*b*) What is object-oriented design? Explain its characteristics. How is it different from function-oriented design ? 10

6. (*a*) Distinguish between top-down and bottom-up integration testing. Which one of these types of integration testing requires stubs and which requires drivers ? Explain your answer. 7

(*b*) What do you understand by path testing ? Explain your answer with an example. 5

(*c*) What is meant by statement coverage, branch coverage and condition coverage criteria ? Show that condition coverage criterion is stronger than branch coverage criterion. 8

7. (*a*) What is McCabe's cyclomatic complexity ? How is it computed ? Illustrate the computation procedure with a suitable example. How is cyclomatic complexity effected, if one of the 'for' loops is replaced by a 'while' loop ? Explain. 10

(*b*) What do you mean by software maintenance ? Why is it needed ? What are the different categories of maintenance ? 6

(*c*) Distinguish between verification and validation. 4

8. (*a*) Explain, with an example, how an intermediate COCOMO provides more accurate estimates as compared to basic COCOMO. 6

(*b*) Briefly explain the SEI CMM software quality model. 8

S'13 : 2 FN : CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** and ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answer may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification.

Figures on the right-hand side margin indicate full marks.

Group A

1. (*a*) What is meant by software project planning ? When does the planning activities begin and end in a software life-cycle ? List the essential activities that a project manager undertakes during project planning. 10

(*b*) How does the use of information hiding help improve the readability of programs ? Illustrate. 5

(*c*) Write a brief note on software risk management activities. 5

2. (*a*) What do you understand by a prototype ? Explain the merits and demerits of prototyping over incremental enhancement of softwares. 8

(*b*) Give an example to illustrate how Petri nets are used for modelling of concurrent processes. 7

(*c*) 'Data stores in a DFD cannot interact directly without a process existing between them'– explain. 5

3. (*a*) What is Software Requirement Specification (SRS) ? Discuss *five* desirable characteristics of a good SRS. 3 + 5

(*b*) Develop the complete SRS document for a Hospital Management System. State all your assumptions. 12

4. (*a*) Define coupling. Discuss different types of coupling. Give examples of each type of coupling. 8

(*b*) What are transform analysis and transaction analysis ? Explain, with a suitable example, how transform analysis is used in structured design. 7

(*c*) Distinguish between recursive and iterative procedures. Illustrate your answer using suitable examples. 5

Group B

5. (*a*) What do you mean by cyclomatic complexities of a program ? How is it computed ? Illustrate the computation using three examples containing (*i*) sequential, (*ii*) if-then-else, and (*iii*) while blocks. How is it used for basis path testing ? 10

(*b*) Distinguish between top-down and bottom-up integration strategies. 4

(*c*) What are meant by statement coverage, edge coverage and condition coverage criteria in white-box testing ? Illustrate. 6

6. (*a*) Explain how intermediate CoCoNo provides better estimation as compared to basic CoCoNo. 10

(*b*) What do you mean by version control ? Why is it considered important in large software project ? 6

(*c*) Distinguish between verification and validation. 4

7. (*a*) What is structured programming ? What are the advantages of using structured programming ? 5

(*b*) What do you understand by abstract data type (ADT) ? How has this feature been used in object-oriented system ? Give example of this. 8

(*c*) What do you mean by side effects of a function call ? Give examples. Discuss the effect of duplicate code in programming. 7

8. (*a*) What is software quality ? Describe the SEI CMM software quality model. 10

(*b*) How is a software failure different from a fault ? Give an example. Hence, define software reliability from probabilistic viewpoint. 5

(*c*) What is meant by software maintenance ? Discuss software maintenance activities. 5

Group C

9. Answer the following in brief: 10 × 2

(*i*) What are meant by milestones and deliverables in the context of software development ?

(*ii*) Distinguish between looseless-coupled and tightly-coupled systems.

(*iii*) How is software different from a program ?

(*iv*) What do you mean by balancing of DFDs ?

(*v*) State Halstead's equations for measuring the size of a software.

(*vi*) Why adding manpower to a late project makes it later ?

(*vii*) What is meant by regression testing of softwares ?

(*viii*) What do you mean by reverse engineering ?

(*ix*) Distinguish between functional cohesion and procedure cohesion.

(*x*) What are the advantages of formally specifying the requirements ?

W'13 : 2 FN : CP 435/EC 425 (1463)

SOFTWARE ENGINEERING

Time : Three hours

Maximum Marks : 100

Answer FIVE *questions, taking* ANY TWO *from* **Group A,** ANY TWO *from* **Group B** and ALL *from* **Group C.**

All parts of a question (a, b, etc.) should be answered at one place.

Answer should be brief and to-the-point and be supplemented with neat sketches. Unnecessary long answer may result in loss of marks.

Any missing or wrong data may be assumed suitably giving proper justification.

Figures on the right-hand side margin indicate full marks.

Group A

1. (*a*) Give an example of a software development project for which the iterative Waterfall model is not suitable. Briefly justify your answer. 7

(*b*) What do you understand by top-down and bottom-up programming ? Explain your answer using a suitable example. 6

(*c*) In a Waterfall model-based software development project, identify five reasons as to why the customer requirements may change after the requirements phase is complete and the SRS document has been signed off. 7

2. (*a*) Do you agree with the following statement ? 'Few, if any, organizations in the real world are structured in purely functional or project format.' Justify your answer. 7

(*b*) What are major advantages of first constructing a working prototype before starting to develop the actual software ? What are the disadvantages of this approach ? 6

(*c*) Explain the main differences between architectural design, high-level design, and detailed design of a software system. 7

3. (*a*) What do you mean by a modular design ? How can you determine whether a given design is modular or not ? Explain your answer. 7

(*b*) Discuss the relative merits of formal and informal requirements specifications. 7

(*c*) What is the difference between requirements analysis and specification ? What is the final outcome of the requirements analysis and specification phase ? 6

4. (*a*) What is the difference between the functional and non-functional requirements of a system ? Identify at least two functional requirements of a bank ATM (Automated Teller Machine) system. Also, identify one non-functional requirement for an ATM system. 10

(*b*) Suppose you are the project manager of a small team developing a business application. Assume that your team has experience in developing several similar products. If you are asked to make a choice between democratic and chief programmer team organizations, which one would you adopt for your team ? Explain the reason behind your answer. 10

Group B

5. (*a*) What do you mean by the term 'software reverse engineering' ? Why is it required ? Explain different activities undertaken during reverse engineering. 10

(*b*) How can you compute the cyclomatic complexity of a program ? Explain your answer using a suitable example. How is cyclomatic complexity useful in program testing ? 10

6. (*a*) Define the terms 'software reliability' and 'software quality'. How can these be measured ? 6

(*b*) Distinguish between software verification and software validation. When during the software life cycle are verification and validation performed ? Can one be used in place of the other ? 8

(*c*) Identify the factors which make the measurement of software reliability a much harder problem than the measurement of hardware reliability. 6

7. (*a*) Give meaningful examples of each of the following types of relationships among classes in an object-oriented program. Only a class diagram and a line of explanation is required in each case : 4 × 3

(*i*) Inheritance

(*ii*) Association

(*iii*) Aggregation

(*iv*) Dependency.

(*b*) Can a program be correct and still not exhibit good quality ? Explain. 8

8. (*a*) Briefly explain how COCOMO can be used for software cost estimation. 8

(*b*) Explain why according to the COCOMO model, when the size of a software is increased by two times, the time to develop the product usually increases by less than two times. 6

(*c*) What do you understand by information hiding in the context of software design ? Explain why a design approach based on the information hiding principle is likely to lead to a re-usable and maintainable design. Illustrate your answer with a suitable example. 6

Group C

9. Answer all the following questions by selecting the most appropriate option. 20 × 1

(*i*) The operation phase in the Waterfall model is a synonym for which one of the following phases ?

(*a*) Coding and unit testing phase

(*b*) Integration and system testing phase

(*c*) Maintenance phase

(*d*) Design phase

(*ii*) Which one of the following phases accounts for the maximum effort during development of a typical commercial software ?

(*a*) Coding

(*b*) Testing

(*c*) Designing

(*d*) Specification

(*iii*) Which one of the following life cycle models lacks the characteristics of iterative software development ?

(*a*) Spiral model

(*b*) Prototyping model

(*c*) Classical Waterfall model

(*d*) Evolutionary model

(*iv*) The radial dimension of the spiral model does not represent the following ?

(*a*) Cost incurred so far

(*b*) Number of features implemented so far

(*c*) Progress in the implementation of the current feature

(*d*) Number of risks that have been resolved so far

(*v*) Effort is measured using which one of the following units ?

(*a*) Persons

(*b*) Person-months

(*c*) Months

(*d*) Rupees

(*vi*) COCOMO estimation model can be used to estimate which one of the following ?

(*a*) LOC

(*b*) Effort

(*c*) Function points

(*d*) Defect density

(*vii*) If a project is already delayed, then adding manpower to complete it at the earliest would be

(*a*) always counter productive.

(*b*) can help to a very limited extent.

(*c*) most effective way to tackle the situation.

(*d*) can cause project completion in the shortest time.

(*viii*) Who, among the following, is a stakeholder in a software development project ?

(*a*) A shareholder of the organization developing the software

(*b*) Any one who is interested in the software

(*c*) Any one who is a source of requirements for the software

(*d*) Any one who might be affected by the software.

(*ix*) Which one of the following is not a goal of requirements analysis ?

(*a*) Weed out ambiguities in the requirements

(*b*) Weed out inconsistencies in the requirements

(*c*) Weed out non-functional requirements

(*d*) Weed out incompleteness in the requirements

(*x*) The modules in a good software design should have the following characteristics :

(*a*) High cohesion, low coupling.

(*b*) Low cohesion, high coupling

(*c*) Low cohesion, low coupling

(*d*) High cohesion, high coupling

(*xi*) Which one of the following is a black box testing approach :

(*a*) Path testing

(*b*) Boundary value testing

(*c*) Mutation testing

(*d*) Branch testing.

(*xii*) Which one of the following can be considered as a program validation technique :

(*a*) Unit testing

(*b*) Integration testing

(*c*) Code review

(*d*) Acceptance testing

(*xiii*) Alpha and Beta testing are considered to be which one of the following types of testing ?

(*a*) Regression testing

(*b*) Unit testing

(*c*) Integration testing

(*d*) System testing

(*xiv*) For a large program, which one of the following integration testing strategy is rarely used :

(*a*) Big-bang

(*b*) Top-down

(*c*) Bottom-up

(*d*) Mixed

(*xv*) Which one of the following is the basic focus of modern quality praradigms ?

(*a*) Process assurance

(*b*) Product assurance

(*c*) Thorough testing

(*d*) Thorough testing and rejection of bad products (*xvi*) Which one of the following statement about cyclomatic complexity metric of a program is fase ?

(*a*) It is a measure of testing difficulty of the program.

(*b*) It is a measure of understanding difficulty of the program.

(*c*) It is a measure of linearly independent paths in the program.

(*d*) It is a measure of size of the program.

(*xvii*) A legacy software product refers to a software that is

(*a*) developed at least 50 years ago.

(*b*) obsolete software product.

(*c*) software product that has poor design structure and code.

(*d*) software product that could not be tested properly before product delivery.

(*xviii*) Which one of the following best characterizes *inheritance* ?

(*a*) It is same as encapsulation

(*b*) Aggregation of information

(*c*) Generalization and specialization

(*d*) Polymorphism

(*xix*) Which one of the following indicates 'Is a kind of' relationship ?

(*a*) Aggregation

(*b*) Association

(*c*) Dependency

(*d*) Inheritance

(*xx*) Which one of the following is a characteristic of a good object-oriented design ?

(*a*) Deep class hierarchy

(*b*) Large number of methods per class

(*c*) Large number of message exchanges per use case

(*d*) Moderate number of methods per class.