(10)

[Knowledge for Development]

KIBABII UNIVERSITY

UNIVERSITY EXAMINATIONS
2015/2016 ACADEMIC YEAR

END OF SEMESTER EXAMINATIONS
YEAR TWO SEMESTER TWO EXAMINATIONS

FOR THE MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

COURSE CODE : MIT 852E

COURSE TITLE : MODELS OF SOFTWARE SYSTEMS

DATE: 14/05/2016 **TIME:200PM-400PM**

INSTRUCTIONS TO CANDIDATES

ANSWER ANY THREE QUESTIONS

Question 1 (20 marks)

a) Define *black box* and *white box* testing. What are the advantages of each approach? Why are both necessary? [6 marks]

b) *Full path coverage* testing requires that every possible *path* through the code be tested at least once. Why is full path coverage testing desirable? For the code fragment above, how many test cases would be needed for full path coverage? Why might full path coverage be impossible to achieve for some programs? [4 marks]

c) *Decision point coverage* testing requires that each outcome of each decision point be tested at least once. Why does this criteria usually require fewer test cases than full path coverage? What kinds of error might this testing miss? [4 marks]

d) Boeing estimates that half the cost of the software development for the 777 aircraft was spent on testing, including regression testing. How might this cost be reduced without sacrificing quality? [6 marks]

Question 2 (20 marks)

a) A software architecture describes a high-level design view of a software system. What are the advantages of explicitly describing the architecture independently from the implementation? [2 marks]

b) Pipe-and-filer architectures treat a task as a series of processing steps, such that the output of one step forms the input of the next step. Give an example of a pipe-and-filter system. What are the advantages and disadvantages of this style? [4 marks]

c) Object oriented architectures can be difficult to develop because each object needs to know which other objects exist and how to call their methods. This limitation can be overcome using implicit invocation. Describe how an implicit invocation scheme works, and give two examples of its use. [4 marks]

d) An *architectural description language (ADL)* is a general purpose language for describing software architectures. What kinds of basic components and basic connectors would you expect an ADL to provide? What advantages are there in using an ADL? [4 marks]

e) The Free Software Foundation proposes to develop an open source suite of tools for developing and debugging Java programs. What architectural styles would you consider for this project, and what aspects of this project would affect your choice of style? [6 marks]

Question 3 (20 marks)

A *data abstraction* describes a data structure and the set of operations that can be performed upon it. What are the advantages of data abstractions? [3 marks]

b) A data abstraction is said to be *adequate* if it provides all the operations on the encapsulated data that other programmers will need. Why is adequacy hard to test? Why does adequacy not guarantee that the abstraction will be convenient to use? [4 marks]

c) Once a data abstraction has been defined, it should be possible to change the implementation of the abstraction without affecting the *correctness* of the rest of the program. What *will* it affect? When should you consider changing the implementation of a data abstraction? [4 marks]

d) What is the difference between *object-oriented design* and *object-oriented programming*? Explain how object oriented designs can be implemented in a procedural programming language. [4 marks]

e) Object oriented programs are claimed to be more maintainable than structured programs. How would you go about testing this claim experimentally? What factors do you think might affect the validity of your experiment? [5 marks]

Question 4 **(20 marks)**

a) Consider the formulas (A) and (B) indicated below:

(A) $(\sim p \vee \sim p)$

(B) $\sim(\sim q \vee \sim(p \& \sim r))$

Draw the truth table for the two formulas and state the logical relationship between them. [4 marks]

b) For the following English sentence identify the connectives, create the dictionary and give the correct formalization using notational logic. [5 marks]

If Luke eats lunch and his lunch is spicy then Luke has a cup of milky tea.

c) Translate the following argument into Predicate Logic formulas and prove its validity.

I. Every decision has both positive aspects and negative aspects.

II. A rational person weighs both positive aspects and negative aspects of each decision.

III. Some person does not weigh positive aspects and negative aspects of any decision.

IV. Therefore, some person is not rational.

[6marks]

d) Give English translations for each of the following assertions and explain how the two are related, assuming P is some predicate. [4 marks]

I. $\forall x \, \exists y \, P(x, y)$

II. $\exists y \, \forall x \, P(x, y)$

Question 5 **(20 marks)**

a) Zave defines Requirements Engineering as *"the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems [and] the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families"*. Why is Requirements Engineering considered to be the most important part of software system development? [5 marks]

b) Requirements should state what a system should do, without stating how it should do it. Why is this distinction useful? [3 marks]

c) *Structured Analysis* proceeds by modelling the current physical system, abstracting a model of the current logical system, and then modelling the new logical system. What are the advantages and disadvantages of building these three separate models? What representations are used for each of these models? [5 marks]

d) Explain why each of the following is an important property of a software specification, and explain how it can be achieved when writing specifications: [7 marks]

   i. validity;
   ii. traceability;
   iii. verifiability.