

kasneb

**CICT PART III SECTION 5**  
**SOFTWARE ENGINEERING**

**WEDNESDAY: 22 May 2019.**

**Time Allowed: 3 hours.**

**Answer ALL questions. Marks allocated to each question are shown at the end of the question.**

**QUESTION ONE**

(a) Software engineers encourage coming up with low coupled modules and classes.

**Required:**

(i) Explain the term “low coupling”. (2 marks)

(ii) Analyse three benefits of low coupled systems in software development. (3 marks)

(iii) Describe three factors that affect software quality. (3 marks)

(b) You have been tasked to come up with a feasibility report for a proposed human resource management software system.

Outline three items that you would capture. (3 marks)

(c) A software engineer concerned with software evolution insists on building software processes that are: flexible, extensible and developed faster.

Justify the above approach using the three key characteristics. (3 marks)

(d) Assess three reasons why system prototyping would be necessary in the above project. (6 marks)

**(Total: 20 marks)**

**QUESTION TWO**

(a) You are required to perform a software requirements engineering task for an organisation whose operations you are not familiar with.

**Required:**

As a software engineer, enumerate four steps that you would follow to execute the above task. (4 marks)

(b) Differentiate between the following software engineering terms:

(i) “Software verification” and “software validation”. (4 marks)

(ii) “Component” and “platform”. (4 marks)

(c) A bibliographic reference is a standard entry which refers a reader to an original source of information cited by an author in the main body of text.

**Required:**

Draw a class diagram for bibliographic references that should minimally include the following concepts:

- An abstract class publication.
- A many to many relationship between Author and Publication entities.
- At least three or more concrete classes refining Publication.
- At least one aggregation relationship. Both ends of the aggregation should also be subclasses of Publication.

(8 marks)

**(Total: 20 marks)**

**QUESTION THREE**

- (a) Describe five types of diagrams that could represent the essentials of a system in unified modelling language (UML). (5 marks)
- (b) Evaluate five characteristics that distinguish embedded systems from other software systems. (5 marks)
- (c) You have been appointed as a project manager of a software system’s project in your company.

**Required:**

- (i) Outline three functions you would be expected to perform. (3 marks)
  - (ii) Enumerate two software project techniques that you would use to estimate the size of the project. (2 marks)
- (d) A mid-sized organisation with various departments requires a software system that would automate its functionalities. The system will be used over a long period of time. It would therefore require proper documentation.

**Required:**

Assess five important areas that the system documentation needs to address to ensure ease of system maintenance in future. (5 marks)

**(Total: 20 marks)**

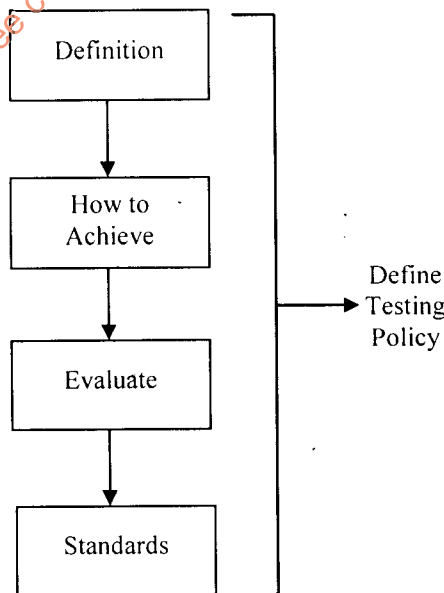
**QUESTION FOUR**

- (a) In relation to software maintenance and evolution, explain the following terms:
  - (i) Change avoidance. (2 marks)
  - (ii) Change tolerance. (2 marks)
- (b) Appraise six tasks that you would undertake when carrying out an application control audit. (6 marks)
- (c) Analyse five best practices in the information technology (IT) budget process. (10 marks)

**(Total: 20 marks)**

**QUESTION FIVE**

- (a) XYZ Company Limited has completed the design and coding of a new system.  
Describe three methods which XYZ Company Limited could use to roll out the new system. (6 marks)
- (b) The figure below shows the important steps used to define a software testing policy.



**Required:**

Discuss in detail how the steps given in the figure above could be used to implement a testing policy of an organisation. (8 marks)

- (c) Traditionally, methodology testing of software development is always done after the build and execution phases.  
With respect to the above statement, write an analysis that cuts across the different phases of software development to justify why this is a wrong approach. (6 marks)

**(Total: 20 marks)**