UGANDA INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

END OF SEMESTER II EXAMINATIONS

DEPARTMENT	:	**ICT**
YEAR	:	**TWO**
SEMESTER	:	**TWO**
PROGRAM	:	**DIPLOMA IN COMPUTER SCIENCE**
COURSE	:	**INTRODUCTION TO SOFTWARE ENGINEERING**
COURSE CODE	:	**CSC2202**
DATE	:	**SATURDAY, JUNE 29, 2024**
TIME	:	**02:00 PM – 05:00 PM**
DURATION	:	**3 Hours**

INSTRUCTIONS

- The paper consists of two Sections A and B
- Answer **ALL** questions in Section A and **ANY THREE** questions in Section B
- Credit will be given to relevant examples and illustrations
- Don Not Write anything on this question paper

SECTION A [40 MARKS]
Attempt ALL the Questions in this Section.
Question 1

A. Computer program and software application are different. Explain. **(2 marks)**

B. The responsibilities of a software engineer are **professional** and **Ethical.**
But they are also **legal** and **social**.
Identify **any one** example for each of the highlighted classification. **(4 marks)**

C. State **one** good quality of a software according to the:
(i) customer (ii) software developer (iii) software company **(6 marks)**

D. (i) Which software size is developed using top-down approach?
(ii) What alternative approach is available for the other size of software? **(2 marks)**

E. (i) Give **any two** examples of complex systems.
(ii) Identify **any two** sub-systems on of these complex systems. **(4 marks)**

F. (i) What are **emergent systems properties**? **(5 marks)**
(ii) Differentiate between software **portabilit**y and **interoperability**. **(2 marks)**

G. System environment can either be **physical** and **organisational**.
Identify **any two** examples of each of these types of environment. **(4 marks)**

H. (i) Which term is used to mean acquiring a system?
(ii) Give the **two main** options available to acquire a system? **(3 marks)**

I. Give a generic model of software development where development starts with:
(i) well understood requirements, then add new features using customers input
(ii) existing components, then integrate them in the software being developed
(iii) poorly understood requirements, then develop a rough software quickly **(3 marks)**

J. (i) What is another name for requirements engineering?
(ii) State the first and last activities in requirements engineering. **(2 marks)**

K. Identify the type of requirements which:
(i) does not contain detailed technical knowledge.
(ii) is also known as functional specification **(3 marks)**

SECTION B [60 MARKS]
Attempt only Three Questions in this Section.
Question 2

(a) Software engineering was as a result of software crisis in 1960s and 1970s.
(i) Identify **any two** issues that constituted the software crisis.
(ii) What mitigation is done to ensure no repeat of each of these issues? **(4 marks)**

(b) (i) What is Systems engineering?
(ii) How has Software engineering benefited from Systems engineering? **(4 marks)**

(c) (i) With examples, explain functional property of a system.
(ii) Differentiate between system safety and security. **(6 marks)**

(d) A system can affect the environment, but also an environment can affect the system. With examples, illustrate these two scenarios. **(4 marks)**

(e) (i) Who is a system analyst?
(ii) What technical knowledge should he / she possess? **(2 marks)**

Question 3

(a) Differentiate between:
(i) software design and implementation
(ii) software verification and validation **(6 marks)**
(iii) system maintainability andevolution

(b) Waterfall development model is the most commonly used for software development.
(i) With the help of a diagram, identify and describe the main activities in the model.
(ii) How differently is it done using Agile development model. **(8 marks)**

(c) Acceptance testing includes alpha testing and beta testing. Briefly explain how each of these software testing is done. **(4 marks)**

(d) (i) Identify **any two** problems of using natural language in software requirements.
(ii) What are the alternatives to natural language for specification? **(4 marks)**

Question 4

(a) (i) What is UML (Unified Modeling Language)?
(ii) Identify **any two** goals of using UML. **(4 marks)**
(iii) Sketch and explain the main sections of UML class diagram.

(b) Differentiate between:
(i) association and multiplicity
(ii) generalisation and inheritance **(4 marks)**

(c) Draw UML Class diagrams showing associations between classes:
Book and Borrower in Library Management System
Student and Course in Academic Registrar Information System **(4 marks)**

(d) In ICT and Engineering Department of UICT:
Each lecturer is allocated 1 to 3 courses.
Each course is allocated to 1 to 4 lecturers.
Each student can take 5 or 6 courses in a semester.
A course has 30 to 80 students.
Required:
Draw UML class diagram with associations and multiplicities. **(8 marks)**

Question 5

(a) The software development team includes; project manager, software engineers and programmers. Identify the main responsibilities of these three categories on the team. **(6 marks)**

(b) What methods does the project manager use to select members of the team?

Identify **any three** factors affecting selection of members of the team. **(6 marks)**

(c) Illustrate with a simplified diagram, how **Abraham Maslow** classified a hierarchy of human needs. Explain using **any three** sections on this diagram what the project manager should do to retain members on his team. **(6 marks)**

(d) Good communication is essential for effective group working.
Give **any two** ways communication is required on a software project team. **(2 marks)**

Question 6

(a) Define these given terms and give the commonly used units for them:
Effort Software size Productivity **(6 marks)**

(b) What is Software cost estimation?
Give **any two** objectives of Software cost estimation?
Identify **any two** parameters that determine of software cost estimation. **(6 marks)**

(c) What is FPA (Function Point Analysis)?
Write in full the five categories, EI, EO, EQ, IF, and ILF of FPA components. **(6 marks)**

(d) Table 1: Shows software cost components with their complexities.

Component Type	**Number of Complexity**		
	Low	**Average**	**High**
EIs	3	4	1
EOs		6	2
EQs		8	
EIFs		3	4
ILFs			5

Table 2: Shows Estimates proposed by **Allan Albrecht in 1979**

Component Catergory	**Weight /Rank**		
	Low	**Average**	**High**
EI	3	4	6
EQ	3	4	6
EO	4	5	7
EIF	5	7	10
ILF	7	10	15

Table 3: Used for computation of Unadjusted Function Point Count (UFP)

Component Category	**Complexity Computations**			**Totals**
	Low	**Average**	**High**	
External Inputs (EIs)				
External Inquiries (EQs)				

External Outputs (EOs)				
External Interface Files (EIFs)				
Internal Logical Files (ILFs)				
Total Unadjusted Function Point Count (UFP)				

Required:
Calculate Unadjusted Function Point Count (UFP) by filling in the table below. **(8 marks)**

END