



PAN AFRICA CHRISTIAN UNIVERSITY

DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY

END OF TERM EXAMINATION

DEPARTMENT: COMPUTING & INFORMATION TECHNOLOGY

COURSE CODE: DICT0212

CAMPUS: ROYSAMBU

COURSE TITLE: SOFTWARE ENGINEERING

EXAM DATE: TUESDAY, 11TH DECEMBER 2018

TIME: 14:00 HRS- 17:00 HRS

INSTRUCTIONS

- This exam script has **TWO (2)** sections.
- Read all questions carefully before attempting.
- Answer ALL questions in Section **A** and any other FOUR questions in Section **B**.
- Write only your **student number** on the answer booklet provided.
- None programmable calculator permitted.
- Calculators on phones, tablets and computers are NOT permitted neither are these device permitted in the Examination in Theory Papers.

SECTION A

(Answer ALL questions in this section)

Question 1:

- I. Define the following terms in the context of software engineering:
 - a. Software Processes (2 Marks)
 - b. Requirements Validation (2 Marks)
 - c. Requirements Verification (2 Marks)
 - d. Project or Product Metrics (2 Marks)
 - e. Seeding in Software Testing (2 Marks)
- II. Discuss the statement "Software Maintenance is Expensive" (2 Marks)
- III. Suppose you have two 'identical & equivalent' software testers Mariga and Conjestina. After they bash away at the application for a while, Mariga finds 15 bugs and Conjestina finds 13. Of the bugs, they find 5 in common. Identify the number of bugs in the application using the following methodologies
 - a. Lincoln Index methodology (2 Marks)
 - b. Seber Estimator. (2 Marks)
- IV. Given two objects, *Student* and *Course* classes, show and draw the class relation indicating that 1 students takes 1 or more courses. (You're free to make any assumptions of the properties, methods and events about the objects). (2 Marks)
- V. You are the Software Architect working on a project to design and deliver an Application for recording the results of an athletic race. Your analyst/programmer presents to you the following table where the table's key is HEAT.

Race (m)	Heat	Time (hrs)	Team	Team	Winner	Time	Time
4x100	1	0900	MIST Tech	REED Oracle	MIST Tech	0:37:04	0:36:84
4x100	2	0920	Software Dojos	Elf Power	Elf Power	0:37:13	0:37:10
4x400	3	0950	Elite Titans	Rainbow Energy	Rainbow Energy	2:55:39	2:55:29
4x400	4	1020	Tech Champs	Jovi Life	Jovi Life	2:56:00	2:54:29

For each of the following normalisations, identify rules the table's design violates?

- a. 1NF. (3 Marks)

- b. 2NF. (1 Marks)
 - VI. Discuss two key goals of software testing. (4 Marks)
 - VII. List four (4) techniques engaged in software testing. (4 Marks)
 - VIII. Discuss four (4) ways in which requirements are recorded (4 Marks)
 - IX. The following describe software process approaches: predictive, iterative, incremental, or agile.
 - a. Identify the approach that gets a working program to users the soonest. (1 Mark)
 - b. Identify the approach that gets a working program to users the Latest. (1 Mark)
 - X. Elaborate on the meaning and/or application of JBGE in coding. (2 Marks)
 - XI. Differentiate between these two (2) categories of Maintenance tasks.
 - a. Perfective and Preventive. (2 Marks)
-

SECTION B

(Answer any FOUR (4) questions in this section)

Question 2:

- I. There exists eight (8) basic tasks that all software engineering projects must handle. Define each of these tasks. (8 Marks)
- II. Discuss seven (7) features that a software engineering document management system should provide. (7 Marks)

Question 3:

- I. FURPS and FURPS+ refer to the acronym used in categorising business, user and systems requirements.
 - a. Define all the categories of FURPS and FURPS+ (9 Marks)
- II. Consider the Waterfall Model and the Unified Process Model software processes.
 - a. Draw a diagram showing the matching of the phases of the waterfall model against those of Unified Process. (2 Marks)
 - b. Explain the main differences. (2 Marks)
- III. In software project management, identify two (2) major mistakes or challenges one has to consider while tracking tasks (2 Marks)

Question 4:

- I. Consider the project application described below:

Your client desires you produce a drawing program named ClassyDraw which allows one to draw line segments, ellipses, polygons, text, and other shapes. ClassyDraw represents each shape

one draws as an object that you can later select, move, resize, modify, and delete.

The initial set of requirement are:

- a. Draw: line segments, sequences of line segments, splines, polygons, ellipses, circles, rectangles, rounded rectangles, stars, images, and other shapes.*
- b. Save and load files.*
- c. Protect the current drawing. For example, if the user tries to close the program while there are unsaved changes, prompt the user.*
- d. Let the user specify the line style and colours used to draw shapes.*
- e. Let the user specify the fill style and colours used to draw shapes.*
- f. Click to select an object.*
- g. Click and drag to select multiple objects.*
- h. Click or click and drag with the Shift key down to add objects to the current selection.*
- i. Click or click and drag with the Ctrl key down to toggle objects in and out of the current selection.*
- j. Click and drag the selected objects to move them.*
- k. Edit the selected objects' line and fill styles.*
- l. Delete the selected objects.*
- m. Select colours from a palette.*
- n. Place custom colours in a custom palette.*
- o. Support transparency.*
- p. Copy and paste the entire drawing, a rectangular selection, or an irregular selection as a bitmapped image.*
- q. Copy, cut, and paste the currently selected objects.*
- r. Allow the user to write scripts to add shapes to a drawing.*
- s. Let the user rearrange the palettes and toolbars.*
- t. Auto-save the current drawing periodically. If the program crashes, allow the user to reload the most recently saved version.*
- u. Auto-save the current drawing every time a change is made. If the program crashes, allow the user to reload the most recently saved version.*
- v. Provide online help.*
- w. Provide online tutorials.*

- a. Using the MuSCoW Methodology,
 - i. Identify the required features necessary for the project to be considered a success. (3 Marks)
 - ii. Identify the features not necessary in this 'first' phase of the project. (2 Marks)
- b. Using high level design, justify the architecture or architectures you consider relevant for the application (4 Marks)

II. Consider the following code extract, focusing on the comments:

```
// Loop through the items in the "items" array.
for (int i = 0; i < items.Length - 1; i++)
{
    // Pick a random spot j in the array.
    int j = rand.Next(i, items.Length);
    // Save item i in a temporary variable.
    int temp = items[i];
}
```

```
// Copy j into i.  
items[i] = items[j];  
// Copy temp into position k.  
items[j] = temp;  
}  
}
```

- a. Explain the problem with comments in the code. (2 Marks)
- b. Re-write the code with corrections applied to the comments. (4 Marks)

Question 5:

- I. Consider the following statements of activities:

In a bakery project, an automated systems is required to receive ingredients to produce bakes cookies.

The initial simple set of activities are described below:

1. *Start oven, mix dry ingredients, and mix wet ingredients; whose steps can all proceed at the same time in parallel.*
2. *When the three parallel activities all are done, combine all the ingredients.*
3. *A test then checks the batter's consistency. If the batter is too sticky, you add more flour and recheck the consistency. You repeat until the batter has the right consistency.*
4. *When the batter is just right, you roll out the cookies, wait until the oven is ready (if it isn't already), and bake the cookies for eight minutes*
5. *After eight minutes, you check the cookies. If the cookies aren't done, you bake them for one more minute. You continue checking and baking for one more minute as long as the cookies are not done.*
6. *When the cookies are ready, you remove them to a cooling rack and stop all the activities.*

- a. Show and draw the UML Activity Diagram detailing the work flows.

(4 Marks)

- II. Suppose you were to issue someone with instructions to drive your car to the nearest supermarket

- a. Using top-down design, write out the highest level of instructions that you would use to instruct someone with. (Keep it at a very high level.)

(4 Marks)

- b. List any assumptions you make.

(2 Marks)

- III. Suppose a software project has a *Counties* table that lists the counties where your Client does business. A search dialogue lets a user select one of the counties from a drop-down list to select accounts from the selected county. Some of the Use Cases call for the county to be set to Taita-Taveta, Makueni, Kwale, and Marsabit, but during tests Kwale doesn't appear in the drop-down list.

- a. Draw an Ishikawa diagram showing possible causes for this problem.

(4 Marks)

- b. Identify any step(s) one could take to try to find the root cause of the problem? (1 Mark)

Question 6:

- I. Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +102 or -21.3e+12 (which means -12.3×10^{12}). Allow both 'E' and 'e' for the exponent symbol. (4 Marks)
- II. List all three (3) steps necessary for a successful software project deployment. (3 Marks)
- III. Having deployed your software project to your client and entered into a maintenance phase, your client regularly reports failures and unwanted behaviour. Your software engineering team is tasked to design a system that keeps track of these bugs.
 - a. Draw a flow-chart showing the movement of the reported bug through the states: *New, Assigned, Reproduced, Cannot Reproduce, Fixed, Tested, and Closed*. Allow the bug to move into *Pending* if it cannot be reproduced. Label the connecting arrows with the tasks that lead to the new state and require approval before moving a bug into the *closed* state. (4 Marks)
 - b. Elaborate the cases in which one would move a bug from the *closed* state to the *deferred* state? (2 Marks)
- IV. Differentiate between High-Level and Low-Level design (2 Marks)

Question 7:

- I. Assume the following scenario:

The following list represents properties of several university-business-oriented classes.

 1. *Student* — Name , Phone , Address , BillingAddress , StudentId
 2. *Monthly* — Name , Phone , Address , FacultyId , MonthlyRate
 3. *Dean* — Name , Phone , Address , FacultyId , Office , Salary , Boss , Employees
 4. *Salaried* — Name , Phone , Address , FacultyId , Office , Salary , Boss
 5. *Supplier* — Name , Phone , Address , Products , SupplierId
 6. *DeputyViceChancellor* — Name , Phone , Address , FacultyId , Office , Salary , Managers

Assume the Supplier is someone who supplies products for the university

 - a. Show and draw an inheritance diagram showing the relationships among these classes. Where necessary indicate any relevant assumptions and/or add extra classes as needed. (7 Marks)
- II. List four (4) tools required for a good development environment (4 Marks)
- III. Differentiate between these levels of testing: (4 Marks)
 - a. Unit testing

**PAN AFRICA CHRISTIAN UNIVERSITY
EXAMINATIONS PAPER**



- b. Component Interface Testing
- c. Acceptance Testing
- d. Destructive Testing