Code: AC16/AT13   **Subject: SOFTWARE ENGINEER**

AMIETE – CS/IT (OLD SCHEME)

Time: 3 Hours   **OCTOBER 2012**   **Max. Marks: 10**

PLEASE WRITE YOUR ROLL NO. AT THE SPACE PROVIDED ON EACH PAGE IMMEDIATELY AFTER RECEIVING THE QUESTION PAPER.

NOTE: There are 9 Questions in all.

- **Question 1 is compulsory and carries 20 marks. Answer to Q.1 must be written in the space provided for it in the answer book supplied and nowhere else.**
- **The answer sheet for the Q.1 will be collected by the invigilator after 45 Minutes of the commencement of the examination.**
- **Out of the remaining EIGHT Questions answer any FIVE Questions. Each question carries 16 marks.**
- **Any required data not explicitly given, may be suitably assumed and stated.**

Q.1   **Choose the correct or the best alternative in the following:** (2×10)

a.   _____ describes the order of tasks and estimates of time and effort attributes.

   (A) Test plan   **(B)** Quality assurance
   (C) Requirements   **(D)** Project schedule

b.   Project risk factor is considered in

   (A) Waterfall model   **(B)** Prototyping model
   (C) Spiral model   **(D)** Interactive enhancement model

c.   Spiral model is an example of

   (A) Evolutionary development model
   (B) Prototype model
   (C) Linear sequential model
   (D) RAD model

d.   In requirement engineering, the term QFD stands for

   (A) Quality function design   **(B)** Quality factor design
   (C) Quality function deployment   **(D)** Quality function development

e.   Number of user inputs, user outputs and user inquiries are used as metrics to compute

   (A) line of count   **(B)** program logic
   (C) function point   **(D)** structure point

f. Which of the following is not a product attribute of an intermediate COCOMO model?

   (**A**) Data size (DATA)
   (**B**) Virtual machine volatility (VIRT)
   (**C**) Product complexity (CPLX)
   (**D**) Required software reliability (RELY)

g. When a module contains tasks that are related by the fact that all must be executed with the same span of time, then the module exhibits _____

   (**A**) Cohesion   (**B**) Logical Cohesion
   (**C**) Temporal Cohesion   (**D**) Coincidental Cohesion 2

h. In basic execution model, the additional time required to reach the failure intensity objective is given by

   (**A**) $\Delta\tau = \dfrac{V_0}{\lambda_0} \ln\left(\dfrac{\lambda_P}{\lambda_F}\right)$ (**B**) $\Delta\tau = \dfrac{V_0}{\lambda_0} \ln\left(\dfrac{\lambda_F}{\lambda_P}\right)$

   (**C**) $\Delta\tau = \dfrac{\lambda_0}{V_0} \ln\left(\dfrac{\lambda_P}{\lambda_F}\right)$ (**D**) $\Delta\tau = \dfrac{\lambda_0}{V_0} \ln\left(\dfrac{\lambda_F}{\lambda_P}\right)$

i. Cyclomatic Complexity is used for _____.

   (**A**) Test analysis
   (**B**) Test planning and test case design
   (**C**) Test documentation
   (**D**) Test maintenance

j. Boehm's model is suited for _____.

   (**A**) Analysis   (**B**) Development
   (**C**) Maintenance   (**D**) Modification

Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.

Q.2 a. Explain various capability maturity models.   **(5)**

   b. What are the strengths and weaknesses of interactive enhancement model over waterfall model and prototype model?   **(5)**

   c. Write short notes on each of the followings: **(6)**

      (i) Design Walkthroughs
      (ii) Critical Design Review

Code: AC16/AT13   **Subject: SOFTWARE ENGINEER**

Q.3 a. Explain crucial process steps in requirement engineering.

b. Using suitable examples explain how followings are used in problem analysis of software requirement analysis:

   (i) Data Flow diagram
   (ii) E – R diagram   **(6)**

c. Explain the importance of requirement reviews in validation process of SRS.   **(5)**

Q.4 a. Describe the following software size metrics:

   (i) Lines of Code (LOC)
   (ii) Function Count   **(5)**

b. Compare the functioning of basic information flow model and sophisticated information flow model.   **(6)**

c. Compute the function point value for a project with the following information domain characteristics:
Number of user inputs = 15, Number of user outputs = 13, Number of user enquiries = 25, Number of files = 12, Number of external interfaces = 4. Assume that all complexity adjustment values and weighting factors (Zij) are set to average value =3. Assume that 14 algorithms have been counted. **(5)**

Q.5 a. Explain the uncertainties in cost estimation models.   **(4)**

b. Explain the functioning of detailed COCOMO model.   **(6)**

c. Write short notes for the following in Putnam Resource Allocation model:
   (i) Productivity versus difficulty
   (ii) Trade-off between time and cost   **(6)**

Q.6 a. Explain the various types of module cohesions and give their respective order of priorities.   **(6)**

b. Give an example to illustrate bottom-up design, top-down design and hybrid design.   **(5)**

c. Explain the features of object oriented design.   **(5)**

Q.7 a. Explain software reliability versus hardware reliability.   **(4)**

b. Compare the following reliability models:

   (i)   Logarithmic Poisson Model
   (ii) Calendar Time component
   (iii) Macro Model   **(9)**

AC16/AT13 / OCTOBER – 2012   3   AMIETE – CS/IT (OLD SCHEME)

c. Draw the graphical plot to illustrate the relation between failure intensity function and mean value function. **(3)**

Q.8 a. Compare functional and structural testing using suitable example. **(6)**

b. Explain any three types of debugging approaches. Give their advantages and disadvantages. **(6)**

c. Explain any **ONE** of the following dynamic testing tools:

(i) Test File & Test Data Generators
(ii) Test Harness & Test Archiving systems **(4)**

Q.9 a. Discuss the problems faced during maintenance of a software. **(6)**

b. A Software project Development Effort of 800 PM (SDE). It is assumed that 10% code will be modified per year (annual change traffic (ACT)). The cost multipliers are given as:
RELY=1.15, ACAP=0.86, AEXP=0.82, LEXP=0.95, DATA=1.08.
Calculate the annual maintenance effort (AME). **(5)**

c. Explain features of dynamic modeling and functional modeling in object oriented software design. **(5)**