**2019(CSE-3111 ODD)**

**Section A**

* 1. Define software engineering. Discuss the scope and necessity of software engineering.
  2. Identity at least four basic characteristics that differentiate a simple program from a software product.
  3. State five symptoms of the present software crisis
  4. What is a software life cycle model? Explain the problems that might be faced by an organization if it does not follow any software life cycle model.
  5. What are the phase entry and phase exit criteria of classical waterfall model?
  6. Identify at least two activities carried out during each quadrant of the spiral model.
  7. Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.
  8. Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:
     1. A system to control anti-lock braking in a car
     2. A virtual reality system to support software maintenance.
     3. A university accounting system that replaces an existing system
     4. An interactive travel planning system that helps users plan journeys with the lowest environmental impact.
  9. Is it possible to combine process model? If so, provide an example.
  10. What is an agile process? Mention some of the principles defined by the agile alliance in their manifesto
  11. How does pair programming work? Write its advantages and disadvantages
  12. In scrum meeting, what are the key questions asked? Who ask the questions to whom?

**Section B**

* 1. What are the three main types of system testing? Explain them.
  2. Briefly describe the software testing life cycle.
  3. What performance tests are carried out to check whether the system needs the nonfunctional requirements identified in the SRS document?
  4. What for software products are required to maintain?\
  5. What are the different types of maintenance that a software product might need? Why are these maintenances required?
  6. What are the different factors upon which software maintenance activities depend?
  7. What do you mean by the term software reverse engineering? Why is it required? Explain the different activities undertaken during reverse engineering.
  8. A novice software engineer has the following beliefs. Give and explain your each of the cases:
     1. Software engineering requires unnecessary documentation, which slows down the project.
     2. Software is flexible; so software requirement changes can be added during any phase of the development process.
     3. If the project time schedule fails, time gap can be reduced by increasing the number of programmers.
  9. Give some examples of known and unpredictable risks.

1. 1. Using your own words, describe the difference between verification and validation. Do both make use of test-case design methods and testing strategies?
   2. Draw control flow graph from the following function and produce test cases to achieve 100%
      1. Statement Coverage
      2. Branch coverage.
      3. Condition Coverage

void printSum(int a, int b){

int result = a + b; if(result > 0)

printcol("red", result); else if(result < 0)

printcol("blue", result); else

do nothing

}

**2018(CSE-3111 ODD)**

**Section A**

* 1. Define Software and Software Engineering.
  2. Is software engineering applicable when WebApps are built? If so, how might it be modified to accommodate the unique characteristics of WebApps?
  3. Many modern applications change frequently before they are presented to the end user and then after the first version has been put into use. Suggest a few ways to build software to stop deterioration due to change.
  4. In Baetjer notes: "The process provides interaction between users and designers, between users and evolving tools, and between designers and evolving tools [technology]." List five questions that (1) designers should ask users, (2) users should ask designers, (3) users should ask themselves about the software product that is to be built, (4) designers should ask themselves about the software product that is to be built and the process that will be used to build it.
  5. What are the advantages and disadvantages of developing software in which quality is "good enough"? That is what happens when we emphasize development speed over product quality?
  6. It is possible to prove that a software component and even an entire program is correct. So why doesn't everyone do this?
  7. Describe agility (for software projects) in your own words.
  8. Most agile process models recommend face-to-face communication. Yet today, members of a software team and their customers may be geographically separated from one another. Do you think this implies that geographical separation is something to avoid? Can you think of ways to overcome this problem?
  9. Why do requirements change so much? After all, don't people know what they want?
  10. Is it possible for any system to be optimized for all software quality attributes? Explain your answer.
  11. Why software standards are important for quality assurance? Explain why a high quality software process should lead to a high quality software product.
  12. What is software metrics? Describe the metrics for object-oriented design.

**Section B**

* 1. Why is it that many software developers don't pay enough attention to requirements engineering? Are there ever circumstances where you can skip it?
  2. Develop a complete use case for the following activity: Searching for books (on a specific topic) using an online bookstore.
  3. What do you think happens when requirement validation uncovers an error? Who is involved in correcting the error?
  4. You have been asked to develop a system that will help with planning large-scaleevents and parties such as weddings, graduation celebrations, birthday parties, etc. Using an activity diagram, model the process context for such a system that shows the activities involved in planning a party (booking a venue, organizing invitations, etc.) and the system elements that may be used at each stage.
  5. Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrollment, so the registration process must include checks that places are available. Assume that the student accesses an electronic course catalog to find out about available courses.
  6. Does “refactoring” mean that you modify the entire design iteratively? If not, what does it mean?
  7. Using your own words, describe the difference between verification and validation. Do both make use of test-case design methods and testing strategies?
  8. Draw control flow graph from the following function and produce test cases to achieve 100%
     + - 1. Statement Coverage
         2. Branch coverage.
         3. Condition Coverage

Read A

Read B

IF A+B > 10 THEN

Print "A+B is Large"

ENDIF

If A > 5 THEN

Print "A Large" ENDIF

* 1. Define Alpha, Beta and Acceptance testing. Draw the block diagram of acceptance testing process.
  2. What are the benefits of involving users in release testing at an early stage in the testing process? Are there disadvantages in user involvement?
  3. Describe the fundamental process of Test-Driven Development (TDD) approach with the help of appropriate figure.

**2017(CSE-3111 ODD)**

**Section A**

1. Define Software and Software Engineering.
2. To help counter terrorism, many countries are planning or have developed computer systems that track large numbers of their citizens and their actions. Clearly this has privacy implications. Discuss the ethics of working on the development of this type of system.
3. Why are large number of legacy software systems written more than 10 years ago still being used today?
4. Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.
5. Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:
   * 1. A system to control anti-lock braking in a car Branch coverage.
     2. A virtual reality system to support software maintenance
     3. A university accounting system that replaces an existing system
     4. An interactive travel planning system that helps users plan journeys with the lowest environmental impact
6. Is it possible to combine process model? If so, provide an example.
7. Explain why test-first development helps the programmer to develop a better understanding of the system requirements. What are the potential difficulties with test-first development?
8. Most agile process models recommend face-to-face communication. Yet today, members of a software team and their customers may be geographically separated from one another. Do you think this implies that geographical separation is something to avoid? Can you think of ways to overcome this problem?
9. To reduce costs and the environmental impact of commuting, your company decides to close a number of offices and to provide support for staff to work from home. However, the senior management who introduce the policy are unware that the developed using agile methods, which rely on close team working and pair programming. Discuss the difficulties that this new policy might cause and how you might get around these problems.
10. What is legacy software? Why should we improve this type of software and how?
11. Write down the steps for level 1, level 2 and level 3 ER-diagram generation processes.
12. What are the differences between the analysis and the data modeling techniques?

**Section B**

1. Develop a sequence diagram showing the interactions involved when a student for a course in a university. Courses may have limited enrollment, so the de checks that places are available. Assume that the course catalog to find out about available courses.
2. Rajshahi City Corporation has decided to develop a Web-based pothole tracking and repair system (PHTRS). A description follows:

Citizens can log onto a website and report the local a website and report the location and severity of potholes. As potholes are reported they are logged within a "city corporation are assigned an identifying number, stored by street address, s umber, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), zone (determined from street address), and repair priority (determined from the size of the pothole).

Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment ass hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used).

Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and taka amount of damage. PHTRS is an online system; all queries are to be made interactively.

Draw a UML use case diagram for the PHTRS system. You'll have to make a number of assumptions about the manner in which a user interacts with this system.

Develop a class model for the PHTRS system.

1. Using your own words, describe the difference between verification and validation. Do both make use of test-case design methods and testing strategies?
2. Draw control flow graph from the following function and produce test cases to achieve 100%
   * 1. A system to control anti-lock braking in a car Branch coverage.
3. Statement Coverage d) Branch coverage. e) Condition Coverage

Void printSum(int a, int b) {

int result = a + b;

if(result > 0)

printcol("red", result);

else if (result < 0)

printcol ("blue", result);

else

do nothing

a) Define software testing. What are the objectives of software testing?

b) What are the principles which must be applied in testing software?

c) Compare the black-box and white-box testing processes.

d) Explain why testing can only detect the presence of errors, not their absence.

a) Quality and reliability are related concepts but are fundamentally different in a number of ways. Discuss the differences.

b) You have been given the responsibility for improving the quality of software across your organization. What is the first thing that you should do? What's next?

c) Can a program be correct and still not be reliable? Explain.

**2016(CSE-3111 EVEN)**

**Section A**

What are the differences between software engineering and computer science?

Software engineering is a layered technology-explain briefly.

Describe the different kind of software myths with examples.

Write the main differences between the prototype and the incremental software development model.

Compare between RAD and spiral software development model.

When we can use the water fall software development model?

What is an agile process? Mention some of the principles defined by the agile alliance in their manifesto.

How does pair programming work? Write its advantages and disadvantages.

In scrum meeting, what are the key questions asked? Who ask the questions to whom?

What is a class diagram in software engineering?

Write the main functionalities of Use-Case diagram.

Distinguish between Use-Case and activity diagram.

**Section B**

Compare between unit testing and integration testing.

Describe briefly the software testing life cycle.

Discuss about high-order testing and smoke testing.

What are the distinguishing features among creational, behavioral and structural design patterns?

Describe MVC framework with figure and mention some of its benefits.

Describe the Brute Force debugging technique.

Write down McCall's quality factors that affect software quality.

What do you mean by quality assurance of computer software?

Describe briefly the elements of software quality assurance.

Describe the software project management activities.

What factors should be considered for building a good software development team.

Describe the software plan structure briefly.

**2016(CSE-3111 ODD)**

**Section A**

1. Define Software engineering. What are the differences between engineering and system engineering?
2. “Software is both a product and a vehicle for delivering a product”- explain it.
3. What are some management myths and realities related to software development?
4. Write the main differences between the prototype and the incremental software development model.
5. Compare between the RAD and spiral software development model.
6. When we can use the water fall software development model.
7. Write the strength and weakness of V-shaped software development model.
8. Write the steps for level 1, level 2, and level 3 ER-diagram generation process.
9. What are the differences between the analysis and the data modeling techniques?
10. What do you mean by requirement analysis in software engineering?
11. Write the main functionalities of Class diagram.
12. Write the differences between the DFD and activity diagram.

**Section B**

1. Compare between unit testing and integration testing.
2. Describe briefly the software testing life cycle.
3. Compare between high-order testing and smoke testing.
4. What do you understand by proactive and reactive risk management strategy?
5. What is risk identification and how is it done?
6. Write down the steps of risk projection.
7. What is software quality? Write the software quality management activities.
8. How to define software quality standards? What are the problems associate with it?
9. Describe the procedure for building the risk table.
10. Describe the software project management activities.
11. What factors should be considered for building a good software development team
12. Describe the software plan structure briefly.