

D (12)

Scoring Indicators

Code: 5132 (15)

| Qn: No | Scoring Indicators | Split score | Total score |
|--------|--|--------------|-------------|
| I 1 | <ul style="list-style-type: none"> Correcting errors that were not discovered during the product development phase. Improving the implementation of the system, and enhancing the functionalities of the system according to the customer's requirements. Porting the software to work in a new environment. <p>Write any two</p> | 1 X 2 | 2 |
| I 2 | Functional requirements specify the expected behavior of the system—which outputs should be produced from the given inputs. They describe the relationship between the input and output of the system. | | 2 |
| I 3 | <p>test case can be considered as comprising a set of test inputs and execution conditions, which are designed to exercise the SUT in a particular manner</p> <p>A group of related test cases that are generally executed together to test some specific behavior or aspect of the SUT is often referred to as a test suite.</p> | 1 X 2 | 2 |
| I 4 | <p>Information hiding can</p> <ul style="list-style-type: none"> reduce the coupling between modules make the system more maintainable effective tool for managing the complexity of developing software <p>Write any 2</p> | 1 X 2 | 2 |
| I 5 | Risk is defined as an exposure to the chance of injury or loss. That is, risk implies that there is a possibility that something negative may happen. | | 2 |
| II 1 | <ul style="list-style-type: none"> Feasibility Study Requirements Analysis and Specification Design Coding and Unit Testing Integration and System Testing Maintenance <p>Explain briefly</p> | 6 (1 X 6) | 6 |
| II 2 | <p>The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. Ie, during the design phase the software architecture is derived from the SRS document. Two distinctly different approaches are available: the traditional design approach and the object-oriented design approach.</p> <p>Traditional design approach Traditional design consists of two different activities; first a</p> | 6 (3 X 2) | 6 |

Scoring Indicators

2

Code: 5132

| | | | |
|-------------|--|-------------------------|----------|
| | <p>structured analysis of the requirements specification is carried out and is followed by a structured design activity. During structured design, the results of structured analysis are transformed into the software design.</p> <p>Object-oriented design approach In this technique, various objects that occur in the problem domain and the solution domain are first identified, and the different relationships that exist among these objects are identified. The object structure is further refined to obtain the detailed design.</p> | | |
| <p>II 3</p> | <div style="text-align: center;"> <p>External Entity Process Output</p> <p>—————>—————</p> <p>Data Flow (a) Data Store</p> </div> <p>The DFD (also known as a bubble chart) is a hierarchical graphical model of a system that shows the different processing activities or functions that the system performs and the data interchange among these functions. Each function is considered as a processing station (or process) that consumes some input data and produces some output data. The system is represented in terms of the input data to the system, various processing carried out on these data, and the output data generated by the system. A DFD model uses a very limited number of primitive to represent the functions performed by a system and the data flow among these functions.</p> | <p>Fig 3 Desc 3</p> | <p>6</p> |
| <p>II 4</p> | <p>Effort and schedule estimates are also required for determining the staffing level for a project during different phases, for the detailed plan, and for project monitoring.</p> <p>The topdown approach utilizes this and considers effort as a function of project size. In this approach, we need to first determine the nature of the function, and then to apply the function, we need to estimate the size of the project for which effort is to be estimated.</p> <p>If productivity is P KLOC/PM, then the effort estimate for the project will be SIZE/P person months. $EFFORT = a * SIZE^b$, where a and b are constants, and project size is generally in KLOC (size could also be in another size measure called function points which can be determined from requirements).</p> | <p>6</p> | <p>6</p> |

Scoring Indicators

Code: 5132

| | | | |
|-------|--|--------------|----|
| | <p>Bottom-up approach.</p> <p>In this approach, the project is first divided into tasks and then estimates for the different tasks of the project are obtained. From the estimates of the different tasks, the overall estimate is determined. That is, the overall estimate of the project is derived from the estimates of its parts.</p> | | |
| II 5 | <p>A program has a static structure as well as a dynamic structure. The static structure is the structure of the text of the program, which is just a linear organization of statements of the program. The dynamic structure of the program is the sequence of statements executed during the execution of the program.</p> <p>The dynamic behavior if the structure in the dynamic behavior resembles the static structure. The closer the correspondence between execution and text structure, the easier the program is to understand</p> <p>The goal of structured programming is to ensure that the static structure and the dynamic structures are the same.</p> | 6 | 6 |
| II 6 | <p>Unit testing is essentially for verification of the code produced by individual programmers, and is typically done by the programmer of the module.</p> <p>Testing Procedural Units</p> <p>when using procedural units for modules, a program can be viewed as a structure chart, in which nodes are functions and edges represent a calling relationship. In unit testing, one, or a small collection, of these modules is to be tested with a set of test cases. As the behavior of a module depends on the value of its parameters as well as the overall state of the system</p> <p>Unit Testing of Classes</p> <p>To test a class, the programmer needs to create an object of that class, take the object to a particular state, invoke a method on it, and then check whether the state of the object is as expected. This sequence has to be executed many times for a method, and has to be performed for all the methods.</p> | 6 (3 X 2) | 6 |
| II 7 | <p>Reviews and testing are two most common QC activities utilized in a project. Whereas reviews are structured, human-oriented processes, testing is the process of executing software (or parts of it) in an attempt to identify defects. The most common approach for quality planning in a project is to specify</p> <p>the QC activities to be performed in the project, and have suitable guidelines for performing each of the QC tasks, such that the chances of meeting the quality goals are high. During project execution, these activities are carried out in accordance with the defined procedures.</p> <p>Explain</p> | 6 | 6 |
| III a | <p>An extreme programming (Agile) project starts with user stories which are short descriptions of what scenarios the customers and users would like the system to support. They are different from</p> | 9 | 15 |