The estimated time for each part is indicated by E.T.

# MCQ Write the letter of the most correct answer [E.T. =30][30 marks]

Part A: Lec1 Product Quality :

1. Determining how easily bugs can be found and ﬁxed considered as …..:

| 1. Flexibility | 1. Testability |
| --- | --- |
| 1. Maintainability | 1. Efficiency |

1. Determining how well the software does what the customer wants considers :

| 1. Correctness | 1. Reliability |
| --- | --- |
| 1. Efficiency | 1. Usability |

1. The way the product should be changed is considered as ….. :

| 1. Revision requirements | 1. Transition requirements |
| --- | --- |
| 1. Operation requirements | 1. None of the above |

1. How to measure the primary SQFs?

| 1. Correctness | 1. Maintainability |
| --- | --- |
| 1. Integrity | 1. All of the above |

1. How the product will be used is what kind of product requirement ?

| 1. Revision requirements | 1. Transition requirements |
| --- | --- |
| 1. Operation requirements | 1. Security requirements |

1. Determining how well the software does what it is supposed to do considers :

| 1. Correctness | 1. Reliability |
| --- | --- |
| 1. Efficiency | 1. Usability |

Part B: Lec2 Measuring system complexity :

1. Measuring how complicated the program code is considered :.

| 1. Complexity | 1. Efficiency |
| --- | --- |
| 1. Testing | 1. None of mention |

1. The complexity measurement method used by counting the number of independent paths is : .

| 1. LOC metric | 1. Black box |
| --- | --- |
| 1. McCabe’s method | 1. White box |

1. Which number of independent paths of the following adds 1 to cyclomatic complexity, start counting from 1:

| 1. (while, do while, for) loops | 1. Variable initialization |
| --- | --- |
| 1. Assign operation | 1. None of the above |

1. Object-Oriented Complexity metrics are

| 1. (DIT) metric | 1. (COB) metric |
| --- | --- |
| 1. (NOC) metric | 1. All of mention |

1. Loop testing technique considered as

| 1. Black box testing | 1. Unique testing technique |
| --- | --- |
| 1. White box testing | 1. Basic path testing technique |

1. Testing technique that allows all possible user-perceived functions to be tested.

| 1. White box | 1. Basic path technique |
| --- | --- |
| 1. Black box | 1. Loop testing technique |

Part C: Lec3 Securing architecture :

1. The broad structure of a software system, that describes its major parts, and how they are put together and interact

| 1. SW architecture | 1. SW engineer |
| --- | --- |
| 1. SW maintenance | 1. Design Patterns |

1. The architecture view that addresses concurrent aspects of the system at run-time, all system processes, start-up and shut-down

| 1. The functional view | 1. The deployment view |
| --- | --- |
| 1. The process view | 1. The logical view |

1. Patterns for solving specific code scenarios

| 1. Design Patterns | 1. Architecture Patterns |
| --- | --- |
| 1. Model view controller | 1. None of the above |

1. Patterns for the skeleton and abstract view of the software

| 1. Design Patterns | 1. Architecture Patterns |
| --- | --- |
| 1. Creational Patterns | 1. None of the above |

1. Design patterns divided mainly into

| 1. Creational Pattern | 1. Structural Pattern |
| --- | --- |
| 1. Behavioral Pattern | 1. All of the above |

1. Which pattern helps to structure the architecture of a system into groups of basic services/functionalities, each at a particular level of abstraction.

| 1. Factory Pattern | 1. Creational Pattern |
| --- | --- |
| 1. Layer Pattern | 1. Structural Pattern |

Part D: Lec4 Securing architecture:

1. A contract that defines all interfaces and the pre- and postconditions:

| 1. Service | 1. Provider |
| --- | --- |
| 1. Consumer | 1. Registry |

1. Software entity that implements the service, it accepts and executes requests from consumers:

| 1. Service | 1. Provider |
| --- | --- |
| 1. Consumer | 1. Registry |

1. Software entity which calls a service provider to request a service:

| 1. Service | 1. Provider |
| --- | --- |
| 1. Consumer | 1. Registry |

1. Software entity, which allows the lookup of services, service providers and their location:

| 1. Service | 1. Provider |
| --- | --- |
| 1. Consumer | 1. Registry |

1. In what language is this code below written in?

| 1. Python | 1. XML |
| --- | --- |
| 1. PHP | 1. JSON |

1. “query = conn.execute("select \* from employees")”, what does this code do ?

| 1. Connects to DataBase | 1. Creates an API |
| --- | --- |
| 1. Executes a query | 1. Fetches first line from the query |