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. Defects per thousand lines of code is measure for …..:

|  |  |
| --- | --- |
| 1. Maintainability | 1. Testability |
| 1. Correctness | 1. Efficiency |

1. Mean time to change is measure for …..:

|  |  |
| --- | --- |
| 1. Maintainability | 1. Testability |
| 1. Correctness | 1. Efficiency |

A web server has been running for a month. From the log ﬁles for that month we see that, of 4000 accesses, 300 attacks were made. Of these, 150 were denial-of-service attacks, of which 42 were successful, 90 were password guessing (of which 35 were successful) and 60 were accidental attacks (caused by errors on the part of the user), of which 20 were successful.

1. The threat probability of denial-of-service or password guessing …..:

|  |  |
| --- | --- |
|  |  |
|  |  |

1. The security probability of accidental attack …..:

|  |  |
| --- | --- |
|  |  |
|  |

1. The threat probability of denial-of-service and accidental attacks …..:

|  |  |
| --- | --- |
|  |  |
|  |  |

Part B: Lec 2 Measuring system complexity:

1. Calculation of cyclomatic complexity:

int number=100;

System.out.print("List of even numbers from 1 to "+number+": ");

for (int i=1; i<=number; i++)

{

if (i%2==0)

{

System.out.print(i + " ");

}

}

|  |  |
| --- | --- |
| 1. 5 | 1. 3 |
| 1. 7 | 1. 9 |

1. which code of the following is more complex?

Code A:

int number=100;

System.out.print("List of even numbers from 1 to "+number+": ");

for (int i=1; i<=number; i++)

{

if (i%2==0)

{

System.out.print(i + " ");

}

}

Code B:

int number=20;

switch(number){

case 10:

System.out.println("10");

break;

case 20:

System.out.println("20");

break;

case 30:

System.out.println("30");

break;

default:

System.out.println("Not in 10, 20 or 30");

}

Code C:

int number = 987654, reverse = 0;

while(number != 0)

{

int remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = number/10;

}

System.out.println("The reverse of the given number is: " + reverse);

Code D:

boolean b1=true;

boolean b2=false;

String s1=String.valueOf(b1);

String s2=String.valueOf(b2);

System.out.println(s1);

System.out.println(s2);

|  |  |
| --- | --- |
| 1. A | 1. B |
| 1. C | 1. D |

1. …………… is used to test that the client’s requirements is done correctly by implementation

|  |  |
| --- | --- |
| 1. Black box testing & White box testing | 1. White box testing |
| 1. Black box testing | 1. None of the above |

1. ……… That determine Cyclomatic complexity of the flow

|  |  |
| --- | --- |
| 1. Black box testing & White box testing | 1. White box testing |
| 1. Black box testing | 1. None of the above |

1. Q5: in ……… , we design test cases by looking at the specification

|  |  |
| --- | --- |
| 1. Black box testing & White box testing | 1. White box testing |
| 1. Black box testing | 1. None of the above |

Part C: Lec3&4 Securing architecture:

1. One of the following is not considered as an architectural view:

|  |  |
| --- | --- |
| 1. Functional view | 1. process view |
| 1. business view | 1. logical view |

1. ….. is an architectural pattern:

|  |  |
| --- | --- |
| 1. MVC | 1. Singleton |
| 1. factory | 1. observer |

1. …. Is an disadvantage of the layers pattern:

|  |  |
| --- | --- |
| 1. reusability | 1. Maintainability |
| 1. performance | 1. ﬂexibility |

1. In a layered architecture, the model in the MVC pattern resides in the ….:

|  |  |
| --- | --- |
| 1. UI layer | 1. presentation layer |
| 1. Data access layer | 1. business layer |

1. what SOA stands for in software engineering:

|  |  |
| --- | --- |
| 1. Software-oriented architecture | 1. Source of Authority |
| 1. System of Authority | 1. Service-oriented architecture |

1. In this line of code app = Flask(\_\_name\_\_), what is \_\_name\_\_?:

|  |  |
| --- | --- |
| 1. Project file name | 1. Function name |
| 1. Web service name | 1. Library name |

1. What is Employees in the following line: api.add\_resource(Employees, '/employees'):

|  |  |
| --- | --- |
| 1. Class | 1. Function name |
| 1. Database name | 1. Database table name |

1. What of the following is not web API method:

|  |  |
| --- | --- |
| 1. post | 1. delete |
| 1. put | 1. update |

1. What does JSON stand for:

|  |  |
| --- | --- |
| 1. Java Security Object Notation | 1. JavaScript Oriented Notation |
| 1. JavaScript Object Notation. | 1. JavaScript Object Name |

1. what is Flask?

|  |  |
| --- | --- |
| 1. A library | 1. A web application framework |
| 1. an application | 1. a built-in function in python |

Part D: Lec3 Secure Coding:

1. ….. attack is the exploitation of the web-session & its mechanism that is usually managed with a session token.:

|  |  |
| --- | --- |
| 1. Session Hacking | 1. Session Hijacking |
| 1. Session Cracking | 1. Session Compromising |

1. Cross-site scripting (XSS) …...:

|  |  |
| --- | --- |
| 1. Injection of commands | 1. Injection of parameters |
| 1. Injection of scripts | 1. All of the above |

1. ….. is a type of attack that tricks site users or administrators to unknowingly perform malicious actions for the attacker like changing order values and product prices, transfer funds from one account to another, change user passwords to hijack accounts.:

|  |  |
| --- | --- |
| 1. SQL Injection | 1. Command Injection |
| 1. Cross-Site Scripting (XSS) | 1. Cross-Site Request Forgery (CSRF) |

1. When people send you phony emails, pop-up messages, social media messages, texts, calls, or links to fake websites in order to hook you into giving out your personal and financial information :

|  |  |
| --- | --- |
| 1. Pharming | 1. Identity Theft |
| 1. Phishing | 1. Password attack |

1. Authorization best practices

|  |  |
| --- | --- |
| 1. Carry API validation | 1. List down all allowable methods |
| 1. Restrict user access towards system-level resources. | 1. All the above |

1. Which statement defines session hijacking most accurately?

|  |
| --- |
| 1. Session hijacking involves stealing a user’s login information and using that information to pose as the user later. |
| 1. Session hijacking involves assuming the role of a user through the compromise of physical tokens such as common access cards. |
| 1. Session hijacking is an attack that aims at stealing a legitimate session and posing as that user while communicating with the web resource or host machine. |
| 1. Session hijacking involves only web applications and is specific to stealing session IDs from compromised cookies. |