Q3] True or False Questions

1. When you test software, you execute a program using real data.	F
2. Testing can reveal the absence of errors NOT their presence.	F
3. Testing is a general concept which includes verification and validation process (V&V).	T
4. In defect testing, a successful test shows that the system operates as intended.	F
5. In validation testing, a successful test is a test that makes the system perform incorrectly and so exposes a defect in the system.	F
6. The main aim of V & V is to establish confidence that the system is 'fit for purpose'.	T
7. Software inspections is concerned with analysis of the static system representation to discover problems.	T
8. Software testing is concerned with exercising and observing product behaviour.	T
9. In software inspection, the system is executed with test data and its operational behaviour is observed.	F
10. Inspection can be used to verify UML design models and software architecture.	T
11. Inspection can be used to verify software prototype.	F
12. Testing can be used to verify software prototype.	T
13. Inspections require execution of a system so should be used after implementation.	F
14. Inspections and testing are opposing verification techniques.	F

15.Inspections can check conformance with the customer's real requirements but not conformance with a specification.	F
16.Inspections can check non-functional characteristics such as performance, usability, etc.	F
17.Unit testing should focus on testing the functionality of objects or methods.	T
18.Component testing should focus on testing component interfaces.	T
19.Unit testing should focus on testing component interactions.	F
20.Unit testing is a defect testing process.	T
21.Inheritance makes it easier to design object class tests.	F
22.The focus in system testing is testing the interactions between components.	T
23.Exhaustive system testing is a possible policy that can be applied.	F
24.In Test-driven development, tests are written before code and 'passing' the tests is the critical driver of development.	T
25.In regression testing, all tests are rerun every time a change is made to the program.	T
26.The primary goal of the release testing process is to convince the supplier of the system that it is good enough for use.	T
27.Software evolution processes depend on the type of software being maintained.	T
28.The proposals for change are the driver for system evolution.	T
29.The first stage of change implementation may involve program understanding, especially if the original system developers are not	

responsible for the change implementation.	T
30.Program evolution dynamics is the study of the processes of system change.	T
31.Handover problem means that the development team have used an agile approach, but the evolution team is familiar with agile methods.	F
32.Software maintenance means modifying a program before it has been put into use.	F
33.Software maintenance is mostly used for changing generic software.	F
34.Maintenance normally involve major changes to the system's architecture.	F
35.In maintenance, the fault repair effort is greater than functionality addition and modification effort.	F
36.The maintenance cost is only affected by technical factors.	F
37.Maintenance corrupts the software structure so makes further maintenance more difficult.	T
38.System reengineering is the process of re-structuring or re-writing part or all of a legacy system while changing its functionality.	F
39.System reengineering is applicable where some but not all sub-systems of a larger system require frequent maintenance.	T
40.Refactoring is the process of making improvements to a program to slow down degradation through change.	T

41. When you refactor a program, you should add new functionality.	F
42. Reengineering is a continuous process of improvement throughout the development and evolution process. refactoring	F
43. Speculative generality occur when the same group of data items (fields in classes, parameters in methods) re-occur in several places in a program. data clumping	F
44. The dependability of a system reflects the user's degree of trust in that system.	T
45. The disruption caused by system failure can't be minimized if the system can be repaired quickly.	F
46. Repairability is not affected by the operating environment.	F
47. Maintainability is a short-term perspective to get the system back into service. repairability	F
48. Repairability is a long-term perspective. maintainability	F
49. Survivability subsumes the ability of a system to continue in operation in spite of component failures.	T
50. User errors should be detected and corrected manually and should not be passed on to the system and cause failures.	T
51. Dependability costs tend to decrease as increasing levels of dependability are required.	F
52. Availability can be defined as the probability that a system, at a point in time, will be operational and able to deliver the requested services.	T
53. if a system is unavailable, then it is unreliable.	T

	الرقم الاكاديمى

54.All systems achieve availability must achieve reliability.	F
55.A system that conforms to its specification may 'fail' from the perspective of system users.	T
56.Faults are a usually a result of system errors that are derived from failures in the system. failures	F
57.Errors do not necessarily lead to system failures.	T
58.Faults do not necessarily result in system errors.	T
59.Removing X% of the faults in a system will improve the reliability by X%. not	F
60.Primary safety-critical systems are embedded software systems whose failure can cause the associated hardware to fail and directly threaten people.	T
61.Secondary safety-critical systems are systems whose failure results in faults in other systems, which can then have safety consequences.	T
62.Accidents are a result of combinations of malfunctions rather than single failures.	T
63.Security is an essential pre-requisite for availability, reliability and safety.	T

Q4] Choose the correct answer:

1. In ………………., the system is tested during development to discover bugs and defects.
    a. [✓ Development testing]
    b. Release testing
    c. User testing
    d. None
2. In ………………., a separate testing team test a complete version of the system before it is released to users.
    a. Development testing
    b. [✓ Release testing]
    c. User testing
    d. None
3. In ………………. , users or potential users of a system test the system in their own environment.
    a. Development testing
    b. Release testing
    c. [✓ User testing]
    d. None
4. In ……………., where individual program object classes are tested.
    a. [✓ Unit testing]
    b. Component testing
    c. System testing
    d. none
5. In ………………., where several individual units are integrated and tested.
    a. Unit testing
    b. [✓ Component testing]
    c. System testing
    d. none
6. In ………………., where some or all of the components are integrated and tested as a whole.
    a. Unit testing

			السكشن
	الرقم الاكاديمى		

b. Component testing
c. System testing
d. none

7. In .... of automated test, you initialize the system with the test case, namely the inputs and expected outputs.
   a. [✓ Setup part]
   b. Call part
   c. Assertion part
   d. none

8. In .... of automated test, you call the object or method to be tested.
   a. Setup part
   b. [✓ Call part]
   c. Assertion part
   d. none

9. In .... of automated test, you compare the result of the call with the expected result.
   a. Setup part
   b. Call part
   c. [✓ Assertion part]
   d. none

10. .... identifies groups of inputs that have common characteristics and should be processed in the same way.
   a. [✓ Partition testing]
   b. Guideline-based testing
   c. A and B
   d. None

11. .... uses previous experience of the kinds of errors that programmers often make when developing components to choose test cases.
   a. Partition testing
   b. [✓ Guideline-based testing]
   c. A and B
   d. None

	الرقم الاكاديمى		السكشن

12.The .... are the data passed from one method or procedure to another.

[✓ a. Parameter interfaces]

b. Shared memory interfaces

c. Procedural interfaces

d. Message passing interfaces

13.The .... are a block of memory that is shared between procedures or functions.

a. Parameter interfaces

[✓ b. Shared memory interfaces]

c. Procedural interfaces

d. Message passing interfaces

14.The .... mean a sub-system encapsulates a set of procedures to be called by other sub-systems.

a. Parameter interfaces

b. Shared memory interfaces

[✓ c. Procedural interfaces]

d. Message passing interfaces

15.The .... mean a sub-system requests services from other sub-system.

a. Parameter interfaces

b. Shared memory interfaces

c. Procedural interfaces

[✓ d. Message passing interfaces]

16.The .... means that a calling component calls another component and makes an error in its use of its interface.

[✓ a. Interface misuse]

b. Interface misunderstanding

c. Timing errors

d. None

17.The .... means that a calling component embeds assumptions about the behaviour of the called component which are incorrect.

a. Interface misuse

[✓ b. Interface misunderstanding]

c. Timing errors

d. None

18.The .... means that the called and the calling component operate at different speeds and out-of-date information is accessed.

a. Interface misuse

b. Interface misunderstanding

c. [✓ Timing errors]

d. None

19.Release testing is usually a .... testing process where tests are only derived from the system specification.

a. [✓ black box]

b. white box

c. A and B

d. None

20.In ...., users of the software work with the development team to test the software at the developer's site.

a. [✓ Alpha testing]

b. Beta testing

c. Acceptance testing

d. None

21.In ...., a release of the software is made available to users to allow them to experiment and to raise problems that they discover with the system developers.

a. Alpha testing

b. [✓ Beta testing]

c. Acceptance testing

d. None

22.In ...., customers test a system to decide whether or not it is ready to be accepted from the system developers and deployed in the customer environment.

a. Alpha testing

b. Beta testing

c. Acceptance testing

d. None

23.…………………… is a stage in a software system's life cycle where software in operational use and is evolving as new requirements are proposed and implemented in the system.

a. Evolution
b. Servicing
c. Phase-out
d. None

24.In …………………… stage, the software remains useful, but the only changes made are those required to keep it operational (i.e. bug fixes).

a. Evolution
b. Servicing
c. Phase-out
d. None

25.In ……………… stage, the software may still be used but no further changes are made to it.

a. Evolution
b. Servicing
c. Phase-out
d. None

26.The ………… law means that a program used in a real-world environment must necessarily change, or else become progressively less useful in that environment.

a. Continuing change
b. Increasing complexity
c. Large program evolution
d. Organizational stability

27.The ………… law means that as an evolving program changes, its structure tends to become more complex. Extra resources must be devoted to preserving and simplifying the structure.

a. Continuing change

b. Increasing complexity
c. Large program evolution
d. Organizational stability

28. The ………… law means that program evolution is a self-regulating process. System attributes such as size, time between releases, and the number of reported errors is approximately invariant for each system release.
a. Continuing change
b. Increasing complexity
c. Large program evolution
d. Organizational stability

29. The ………… law means that over a program's lifetime, its rate of development is approximately constant and independent of the resources devoted to system development.
a. Continuing change
b. Increasing complexity
c. Large program evolution
d. Organizational stability

30. The ………… law means that over the lifetime of a system, the incremental change in each release is approximately constant.
a. Conservation of familiarity
b. Continuing growth
c. Declining quality
d. Feedback system

31. The ………… law means that the functionality offered by systems has to continually increase to maintain user satisfaction.
a. Conservation of familiarity
b. Continuing growth
c. Declining quality
d. Feedback system

32. The ………… law means that the quality of systems will decline unless they are modified to reflect changes in their operational environment.
a. Conservation of familiarity

	الرقم الأكاديمي		اسم الطالب
			السكشن

b. Continuing growth
c. [✓ Declining quality]
d. Feedback system

33.The .... law means that evolution processes incorporate multiagent, multiloop feedback systems and you have to treat them as feedback systems to achieve significant product improvement.
a. Conservation of familiarity
b. Continuing growth
c. Declining quality
d. [✓ Feedback system]

34.In reengineering process, the activity .... convert code to a new language.
a. [✓ Source code translation]
b. Reverse engineering
c. Program structure improvement
d. Program modularization

35.In reengineering process, the activity .... analyzes the program to understand it.
a. [✓ Reverse engineering]
b. Program structure improvement
c. Program modularization
d. Data reengineering

36.In reengineering process, the activity .... restructure program automatically for understandability.
a. Reverse engineering
b. [✓ Program structure improvement]
c. Program modularization
d. Data reengineering

37.In reengineering process, the activity .... reorganize the program structure.
a. Source code translation
b. Program structure improvement

	الرقم الاكاديمى		

c. Program modularization

d. Data reengineering

38.In reengineering process, the activity .... clean-up and restructure system data.

a. Source code translation
b. Reverse engineering
c. Program structure improvement
d. Data reengineering

39.In legacy system, the .... systems should be scrapped.

a. Low quality, low business value
b. Low-quality, high-business value
c. High-quality, low-business value
d. High-quality, high business value

40.In legacy system, the .... systems should be re-engineered or replaced if a suitable system is available.

a. Low quality, low business value
b. Low-quality, high-business value
c. High-quality, low-business value
d. High-quality, high business value

41.In legacy system, the .... systems should be replaced with COTS, scrapped completely or maintain.

a. Low quality, low business value
b. Low-quality, high-business value
c. High-quality, low-business value
d. High-quality, high business value

42.In legacy system, the .... systems should continue in operation using normal system maintenance.

a. Low quality, low business value
b. Low-quality, high-business value
c. High-quality, low-business value
d. High-quality, high business value

	الرقم الاكاديمى		السكشن

43..................... happens because of design and manufacturing errors or because components have reached the end of their natural life.

a. [✓ Hardware failure]
b. Software failure
c. Operational failure
d. None

44..................... happens due to errors in its specification, design or implementation.

a. Hardware failure
b. [✓ Software failure]
c. Operational failure
d. None

45..................... happens because mistakes made by human operators.

a. Hardware failure
b. Software failure
c. [✓ Operational failure]
d. None

46.............. is a human behavior that results in the introduction of faults into a system.

a. [✓ Human error or mistake] √
b. System fault
c. System error
d. System failure

47.................. is a characteristic of a software system that can lead to a system error.

a. Human error or mistake
b. [✓ System fault] √
c. System error
d. System failure

48.................. is an erroneous system state that can lead to system behavior that is unexpected by system users.

a. Human error or mistake

أسم الطالب			
السكشن		الرقم الاكاديمى	

b. System fault
c. [✓ System error]
d. System failure

49.................. is an event that occurs at some point in time when the system does not deliver a service as expected by its users.

a. Human error or mistake
b. System fault
c. System error
d. [✓ System failure]