Q15. The traditional life cycle for the information system development model is known as \_\_\_\_\_\_\_\_\_\_\_.

1. Waterfall life cycle model
2. Incremental development model
3. Unified software development model
4. Prototyping model

**Answer: A**

Q16. Which of the following is/are the deliverable of the system engineering phase?

1. High-level architectural specification
2. Software architecture specification
3. Functional specification
4. Design specification

**Answer: A**

Q17. In which phase various fact-finding techniques are used?

1. System Engineering
2. Requirement analysis
3. Maintenance
4. Testing

**Answer: B**

Q18. Which of the following is or are disadvantages of the traditional life cycle (TLC)?

1. Activities can not be repeated easily
2. Unresponsive to the change to the client requirements
3. A simple sequential life cycle model and activities do not overlap
4. Each phase has defined deliverables

**Answer: A, B, C [D is an advantage]**

Q19. In software development a prototype is a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1. Partially completed system to explore some aspect of the systems’ requirement
2. Tested and final system
3. System for testing and discarded after testing
4. None of the above

**Answer: A**

Q20. Which of the following can be purpose of construction a prototype?

1. to explore some aspect of the systems’ requirement
2. to determine whether a particular implementation platform can support certain processing requirement
3. the feasibility and usefulness of the system can be tested, even though, by its very nature, the prototype is incomplete
4. to analyze the user requirements easily and test for errors early

**Answer: A, B, C**

Q21. A prototype is intended as the final working system.

Do you agree?

1. Yes
2. No

Answer: B

Q22. Through which of the following ways, users can be involved in an information system development project?

1. As part of the development team
2. In fact gathering
3. Via a consultative approach
4. As a interface designer

Answer: A, B, C

Q23. Which one is **Upper-CASE** tool?

1. A CASE tool that provide support for the analysis and design
2. A CASE too that that provides support for the construction and maintenance of software

Answer: A

Q24. Which one is **Lower-CASE** tool?

1. A CASE tool that provide support for the analysis and design
2. A CASE too that that provides support for the construction and maintenance of software

Answer: B

Q23. Objects communicate each other by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Which one best fits the blank.

1. By sending messages
2. By creating in-memory connection
3. By implementing inheritance
4. By using collaborator

**Answer: A**

Q24. Objects communicate by \_\_\_\_\_\_\_\_\_\_\_.

1. sending message
2. generating event
3. sending signals
4. encapsulating each other

**Answer: A**

Q25. An object encapsulates data and processes to act on this data. These processes are called \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Which one best fits the blank.

1. Signature
2. Protocol
3. Operations
4. State

**Answer: C**

Q26. Each operation has a specific \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Which one best fits the blank.

1. Structure
2. Argument
3. Parameter
4. Signature

**Answer: D**

Q27. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is the definition of an object’s interface.

Which one best fits the blank.

1. An Operation Signature
2. An attribute
3. A method
4. A message

**Answer: A**

Q28. In order to invoke an operation, its \_\_\_\_\_\_\_\_\_ must be given.

Which one best fits the blank.

1. Structure
2. Argument
3. Parameter
4. Signature

**Answer: D**

Q29. Operation signatures are also called \_\_\_\_\_\_\_\_\_\_.

Which one best fits the blank.

1. Message binding
2. Message protocols
3. Message calling
4. Asynchronous operation

**Answer: B**

Q30. Which of the following is or are true about polymorphism?

1. Polymorphism allows one message to be sent to objects of different classes and each object responds differently
2. Polymorphism allows to create more than one object of a class in the same operation
3. Polymorphism allows to restrict access to an object’s operations
4. All of the above

**Answer: A**

Q31. “The ability of different methods to implement the same operation in different ways those are appropriate to its class”

This statement is about \_\_\_\_\_\_\_\_\_.

1. Inheritance
2. Generalization
3. Specialization
4. Polymorphism

Answer: D

Q32. Which of the following is or are true?

1. An object’s state is determined by the values of its attributes
2. An object maintains its state until an external stimulus change it
3. An object’s state affects the way it responds to messages
4. An objects state can not be changed by its own operation

Answer: A, B, C

**Question 10**

One of the major challenges during system installation is which of the following?

1. spacerEnsuring that the new software is correctly installed to use the computer effectively.
2. **Avoiding unnecessary disruption and minimizing the attendant risk of change.**
3. Ensuring that both old and new systems run in parallel.

**Question 11**

Which of following is true about software construction in the traditional life cycle?

1. spacerOnly one programming language could be used.
2. Relational database management systems are not used.
3. **The design is used to develop program code.**

**Question 12**

Which of the following is a disadvantage of the traditional life cycle?

1. spacerIt does not allow the use of object-oriented technology.
2. **Requirements change during development after the main system requirements have been agreed and are difficult to accommodate.**
3. It separates requirements analysis and design.

**Question 13**

Iteration is problematic during the traditional life cycle for which of the following reasons?

1. spacerArchitectural decisions are difficult to change.
2. **Ad hoc coding solutions may be used to address changes in requirements**
3. Requirements will change during the project.

**Question 14**

Which of the following statements is true about a prototype system?

1. spacerA prototype system is always discarded before the final production system is built.
2. Rapid development tools are only used to build prototype systems.
3. **A prototype system is incomplete or lacks the resilient construction of the final production system.**

**Question 15**

Which of the following is not an advantage of prototyping?

1. spacerPrototyping is easy to manage.
2. Prototypes may be used to reduce misunderstandings about requirements.
3. **Prototyping requires no analysis or design.**

**Question 16**

Which of the following is not a workflow in the Unified Software Development Process?

1. **spacerConstruction**
2. Implementation
3. Test

**Question 17**

User involvement in software development is important for which of the following reasons?

1. spacerIt is cheaper to have users as part of the project team rather than professional software developers.
2. Users understand why the requirements cannot be met.
3. **Users can influence the way a project proceeds by identifying the most acceptable course of action from various alternatives.**

**Question 18**

Consider the following statements about CASE tools:

Current CASE tools can perform semantic checks on a set of diagrams modelling an information system.  
Current CASE tools can perform syntactic and consistency checks on a set of diagrams modelling information system.  
Current CASE tools can perform syntactic checks on a set of diagrams modelling information system.

Which of the following is true?

1. Statements A, Band C are true.
2. Statements A and C are true.
3. **Statements B and C are true.**

**Question 19**

Which of the following is an example of a systems development methodology?

1. spacerThe traditional life cycle.
2. The Unified Modeling Language.
3. **The Unified Software Development Process.**

**Question 20**

What are the key distinguishing features of an agile method?

1. spacerThey allow development staff to move freely from one project to another.
2. **They are documentation light and are responsive to changes in user requirements.**
3. They produce flexible systems that are easy to change.

spacer

**Question 1:** Which of the following best describes an object?

1. Part of a software system that is entirely unique.
2. **A concept, abstraction or thing in an application domain.**
3. A program that represents something tangible in the problem domain.

**Question 2:** Which of the following best describes abstraction?

1. A representation of something tangible.
2. A representation that can be stored in a software system.
3. **A representation that contains only relevant details.**

**Question 3:** Which of the following is not a reason for modelling objects?

1. To produce a design for part of a software system.
2. To understand an aspect of the application domain.
3. **To separate data from process.**

**Question 4:** What do all objects have?

1. **State, behaviour and identity.**
2. Behaviour, data and identity.
3. Instances, structure and similarity.

**Question 5:** Which of the following best describes object state?

1. **The particular condition that an object is in at a given moment, determining its possible behaviours.**
2. Which class the object belongs to.
3. The semantics of the object.

**Question 6:** Which of the following best describes object behaviour?

1. What the object is able to do to other objects.
2. **What the object is able to do for other objects.**
3. What the object is able to do to itself.

**Question 7:** Which of the following is a useful set of questions to ask when modelling an object, according to Rebecca Wirfs-Brock?

1. **Who am I, what can I do and what do I know?**
2. Where am I, what am I and who do I know?
3. What do I have, what can I get and what can I do?

**Question 8:** Which of the following is not a description of a class?

1. A set of objects that share the same behaviour, attributes, relationships and semantics.
2. An abstract descriptor for a set of instances with certain logical similarities to each other.
3. **A set of objects that collaborate together to achieve some common objective.**

**Question 9:** Which of the following best describes the relationship between an object and its class?

1. **The structure and permitted behaviours of an object are defined by its class.**
2. A class is a container that holds a collection of similar objects.
3. An object is an implementation of a class.

**Question 10:** What is generalization?

1. A process of broadening the scope of an object, such that it becomes more generally useful.
2. **A kind of relationship between a more general element and a more specific element.**
3. A process of collecting together objects into their respective classes.

**Question 11:** Which of the following best describes a type?

1. **A description of a set of objects with similar behaviours.**
2. A superclass in a generalization hierarchy.
3. A class with a characteristic that distinguishes it from all other classes.

**Question 12:** Which of the following is not an advantage of using generalization?

1. Generalization helps to organize a model so that the degree of similarity between classes is made more explicit.
2. A generalization hierarchy is easy to extend to fit a changing picture.
3. **Generalization helps to encapsulate classes and subsystems so that their implementation is hidden from other parts of the system.**

**Question 13:** How does generalization differ from inheritance?

1. It doesn't - they are the same thing.
2. **Inheritance is a mechanism by which some OO languages implement generalization.**
3. With generalization each class has only one superclass, whereas with inheritance each class has two or more superclasses.

**Question 14:** Which of the following is **not** a characteristic of a subclass?

1. **A subclass can only have superclasses, it cannot have subclasses of its own.**
2. A subclass inherits all the characteristics of its superclass.
3. A subclass includes at least one detail that is not shared by its superclass.

**Question 15:** What is meant by 'transitive operation' in the context of generalization and inheritance?

1. An operation in a superclass may be overwritten by a different operation in a subclass.
2. An operation in a superclass may not be overwritten by a different operation in a subclass.
3. **A subclass inherits characteristics from all its superclasses at all levels.**

**Question 16:** What is the significance of message-passing in an OO system?

1. Messages represent input from users that tells the software system what to do.
2. **Objects exchange messages in order to communicate with each other.**
3. Messages represent output to users that show the results of processing.

**Question 17:** What is a message protocol or signature?

1. A message protocol is a valid sequence of keystrokes by a user.
2. A message protocol is a valid sequence of operations in a series of different objects.
3. **A message protocol is the interface to an operation.**

**Question 18:** What is meant by multiple inheritance?

1. **Multiple inheritance signifies that a class simultaneously belongs to more than one generalization hierarchy.**
2. Multiple inheritance signifies that a class has more than one superclass.
3. Multiple inheritance signifies that a class can have different superclasses at different times.

**Question 19:** Which of the following best describes encapsulation?

1. The implementation of an object can only be changed by its original programmer.
2. **Data within an object can only be accessed by passing a valid message to one of its own operations.**
3. Data within an object can only be accessed by passing a valid message to its class.

**Question 20:** Which of the following best describes an object's interface?

1. The view that an object presents to users of the system.
2. The links that an object has with other objects.
3. **The complete set of signatures for all the object's operations.**

**Question 21:** Which of the following best describes polymorphism?

1. The capacity of an object to behave in different ways at different times according to its current state.
2. **The capacity of different objects to respond to a similar message in appropriate but different ways.**
3. The capacity of an object to send different messages to different objects according to their class.

**Question 22:** Which of the following is a valid reason why it is difficult to design event-driven software in a procedural manner?

1. **It is difficult to anticipate and design for all possible sequences of use.**
2. Procedurally designed programs are not capable of responding quickly to events.
3. Procedural programs are only suitable for record-based data structures.

**Question 23:** Which of the following is not an advantage of modular software design?

1. Modular systems are typically more reliable in use.
2. Modular systems can be implemented in small, manageable chunks.
3. **Modular systems are independent of the operating system that they run on.**