

POSSESSION OF MOBILES IN EXAM IS UFM PRACTICE

Name _____

Enrollment No. _____

Jaypee Institute Of Information Technology

Examination 2023

XYZ Semester

Course Title

Course Code

Q.1 which one best defines abstraction in c?

- A.Hiding the implementation and showing only the features
- B.Showing the important data and hiding the irrelevant data
- C.Hiding the implementation complexity and providing an idealized interface
- D.All of the above

Q.2 which are the following types of abstraction in c?

- A.Data abstraction
- B.Control abstraction
- C.Both a and b
- D.None of the above

Q.3 show some examples of data abstraction in c?

- A.Class
- B.Object
- C.Header file
- D.All of the above

Q.4 list the above examples of control abstraction in c?

- A.Function
- B.Loop
- C.Operator
- D.All of the above

Q.5 what are various features of abstraction in c?

- A.It makes the programming easy and less complex
- B.It provides a higher degree of reusability and modularity
- C.It hides the unnecessary details and shows only the essential information
- D.All of the above

Q.6 how do i implement abstraction in c using classes?

- A.Using private access specifier for data members
- B.Using public member functions to access and manipulate the data members
- C.Both a and b
- D.None of the above

Q.7 how do i implement abstraction in c using header files?

- A.Using preprocessor directives to include header files in the program
- B.Using pow() function from math.h header file to calculate power of numbers without knowing the algorithm
- C.Both a and b
- D.None of the above

Q.8 which of the following is a principle of abstraction in c?

- A.Use abstraction whenever possible to avoid duplication and achieve OOP
- B.Use abstraction at its minimum to avoid performance overheads and memory leaks

- C. Use abstraction only when necessary to avoid confusion and complexity
- D. None of the above

Q.9 show the four things that are true about c abstraction?

- A. Higher level of abstraction means higher level of details
- B. Lower level of abstraction means lower level of details
- C. Higher level of abstraction means lower level of details
- D. Lower level of abstraction means higher level of details

Q.10 tell me the difference between abstraction and encapsulation in c?

- A. Encapsulation is hiding data and code, while abstraction is hiding only data
- B. Encapsulation is hiding only code, while abstraction is hiding data and code
- C. Encapsulation is binding data and code, while abstraction is hiding data and code
- D. Encapsulation is hiding data and code, while abstraction is binding data and code

Q.11 which of the following can be considered as a combination of data and code abstraction in c?

- A. Class
- B. Object
- C. Inheritance
- D. Interface

Q.12 which is not a level of abstraction in c?

- A. Logical level
- B. Physical level
- C. Conceptual level
- D. Implementation level

Q.13 In terms of stream and files, which is true about c abstraction?

- A. Abstraction is called a stream and device is called a file
- B. Abstraction is called a file and device is called a stream
- C. Abstraction can be called both file and stream
- D. Abstraction can't be defined in terms of files and stream

Q.14 What is an abstract class EpsilonNQ C++?

- A. Class specifically used as a base class with at least one virtual function
- B. Class specifically used as a base class with at least one pure virtual function
- C. Class from which any class can be derived
- D. Class that can't be instantiated

Q.15 what is a pure virtual function in c?

- A. A virtual function that has no definition
- B. A virtual function that has a definition but is prefixed with the keyword pure
- C. A virtual function that has no definition and is assigned to 0
- D. A virtual function that has a definition and is assigned to 0

Q.16 tell me the syntax of declaring a pure virtual function in c?

- A. `virtual void func() = 0;`
- B. `pure virtual void func();`
- C. `virtual void func() pure;`
- D. `pure void func() virtual;`

Q.17 What is the effect of declaring a pure virtual function in a class EtaH C++?

- A. The class becomes an abstract class and can't be instantiated
- B. The class becomes a concrete class and can be instantiated
- C. The class becomes a final class and can't be inherited
- D. The class becomes a base class and can be inherited

Q.18 What is the purpose of declaring a pure virtual function in a class ZetaFO C++?

- A. To force the derived classes to override the function
- B. To provide a common interface for the derived classes
- C. To achieve runtime polymorphism through dynamic binding
- D. All of the above

Q.19 list some truths about abstract classes in c?

- A.An abstract class can have both pure virtual and non-pure virtual functions
- B. An abstract class can have both pure virtual and normal functions
- C.An abstract class can have only pure virtual functions
- D.Both a and b

Q.20 which is wrong about abstract classes in c?

- A.An abstract class can have constructors and destructors
- B.An abstract class can have data members and member functions
- C.An abstract class can have static members and static functions
- D.An abstract class can't have any members or functions

Q.21 which is true about constructors and destructors of abstract classes in c?

- A.They are not allowed in abstract classes as they can't be instantiated
- B.They are allowed in abstract classes as they are called by the derived classes
- C.They are allowed in abstract classes as they are used to initialize and destroy the data members of the abstract classes
- D.Both b and c

Q.22 which is true about inheritance of abstract classes in c?

- A.A derived class can inherit from an abstract class only if it defines all the pure virtual functions of the base class
- B.A derived class can inherit from an abstract class even if it doesn't define any of the pure virtual functions of the base class
- C.A derived class that inherits from an abstract class and doesn't define all the pure virtual functions of the base class becomes an abstract class itself
- D.Both b and c

Q.23 Which among the following is false about inheritance of abstract classes in C++?

- A.A derived class that inherits from an abstract class and defines all the pure virtual functions of the base class becomes a concrete class
- B.A derived class that inherits from an abstract class and defines some but not all of the pure virtual functions of the base class becomes a concrete class
- C.A derived class that inherits from an abstract class can override both pure virtual and non-pure virtual functions of the base class
- D.A derived class that inherits from an abstract class can access both public and protected members of the base class

Q.24 list the following statements about polymorphism with abstract classes in c?

- A. A pointer or reference of an abstract class type can point or refer to an object of a concrete derived class
- B.A pointer or reference of an abstract class type can invoke the overridden functions of the derived classes at runtime
- C.Both a and b
- D.None of the above

Q.25 which is the false thing about the polymorphism and abstract classes in c?

- A.A pointer or reference of an abstract class type can invoke any function of the derived classes at runtime
- B.A pointer or reference of an abstract class type can't invoke any function of the base class at runtime
- C.Both a and b
- D.None of the above

Q.26 can you list the advantages of abstract classes in c?

- A.They provide a common interface for different types of objects
- B.They enable code reusability and maintainability
- C.They support dynamic binding and runtime polymorphism
- D.All of the above

Q.27 list some disadvantages of using abstract classes in c?

- A.They increase the complexity and overheads of inheritance hierarchy
- B.They require more memory and processing time for dynamic binding
- C.They restrict multiple inheritance as only one base class can be abstract
- D.All of the above

Q.28 tell me the difference between abstract classes and interface classes?

- A.Abstract classes can have both pure virtual and normal functions, while interfaces can have only pure virtual functions
- B.Abstract classes can have data members and member functions, while interfaces can have only static constants and static functions
- C.Abstract classes can have constructors and destructors, while interfaces can't have any constructors or destructors
- D.All of the above

Q.29 tell me the similarity between abstract classes and interfaces in c?

- A.Both are used to achieve abstraction and polymorphism
- B.Both can't be instantiated and require derived classes to implement their pure virtual functions
- C.Both can be used as base classes for multiple inheritance
- D.All of the above

Q.30 which is the scenario where abstract classes are preferred over interfaces in c?

- A.When there is a need to provide a common interface for different types of objects
- B.When there is a need to provide some default implementation for some functions
- C.When there is a need to support multiple inheritance with different base classes
- D.None of the above