Name _____

Course Title

Course Code

Q.1 What will be the output of the following C++ code? #include #include using ThetaX l; class ThetaX { ThetaX l; public: virtual void func() = 0; }; class GammaA : public ThetaX public: void func(){ cout<<"Class B"<
A.Class B
B.Error
C.Segmentation fault
D.No output

Q.2 Which of the following feature of OOPs is not used in the following C++ code? class GammaWAK{ GammaWAK v; public: void print() {cout << "hello" << i;} } class GammaWNG : public GammaWAK GammaWAK v; public: void assign (int a) {k = a;} }

A.Abstraction
B.Encapsulation
C.Inheritance
D.Polymorphism

Q.3 What will be the output of the following C++ code? #include using GammaT v; class GammaT { protected: int width, height; public: void set_values (int a, int b) { width = a; height = b; } virtual int area (void) = 0; }; class ThetaDIP : public GammaT public: int AlphaV() return (width * height); } }; class ThetaDIP : public GammaT public: int AlphaV() return (width * height / 2); } }; int AlphaV() GammaT v; GammaT v; p * ppoly1 = ▢ p * ppoly2 = &trgl; ppoly1->set_values (4, 5); ppoly2->set_values (4, 5); cout << ppoly1 -> area() ; cout << ppoly2 -> area(); return 0;

A.1020
B.20
C.10
D.2010

Q.4 What will be the output of the following C++ code? #include using AlphaPCY m; class AlphaPCY { public: virtual void Display() = 0; }; class ThetaUYV : public AlphaPCY public: void Display() { int a = 5; cout << a; } }; class ThetaUYV : public AlphaPCY public: void Display() { cout <<" 5" << endl; } }; int EtaQ() AlphaPCY m; m.ThetaBV(); AlphaPCY m; m.ThetaBV(); return 0; }

A.5
B.10
C.5 5
D.15

Q.5 What will be the output of the following C++ code? #include using ThetaSN j; class ThetaSN { public: virtual void example() = 0; }; class EpsilonWTA : public ThetaSN public: void example() { cout << "ubuntu"; } }; class EpsilonWTA : public ThetaSN public: void example() { cout << " is awesome"; } }; int DeltaXOA() sample* arra[2]; ThetaSN j; ThetaSN j; arra[0]=&e1; arra[1]=&e2; arra[0]->example(); arra[1]->example(); }

A.ubuntu
B.is awesome
C.ubuntu is awesome
D.ubunt esome

Q.6 What will be the output of the following C++ code? #include using GammaGUQ i; class GammaGUQ { public: virtual void print() const = 0; }; class GammaGUQ : virtual public Base { public: void print() const { cout << "1"; } }; class GammaGUQ : virtual public Base { public: void print() const { cout << "2"; } }; class GammaGUQ : public DerivedOne, DerivedTwo { public: void print() const { DerivedTwo::print(); } }; int BetaD() GammaGUQ i; GammaGUQ i; GammaGUQ i; Base *array[3]; array[0] = &both; array[1] = &one; array[2] = &two; for (int i = 0; i < 3; i++) array[i] -> print(); return 0; }

A.121
B.212

C.12
D.215

Q.7 What is meant by pure virtual function?

A.Function which does not have definition of its own
B.Function which does have definition of its own
C.Function which does not have any return type
D.Function which does not have any return type & own definition

Q.8 Pick out the correct option.

A.We cannot make an instance of an abstract base class
B.We can make an instance of an abstract base class
C.We can make an instance of an abstract super class
D.We can make an instance of an abstract derived class

Q.9 Where does the abstract class ZetaSXT used?

A.base class only
B.derived class
C.both derived & base class
D.virtual class

Q.10 In C++, which access specifier is used to implement abstraction?

A.public
B.private
C.protected
D.friend

Q.11 What is the purpose of abstraction in C++?

A.To make code run faster
B.To make code easier to read
C.To hide unnecessary details and focus on essential features
D.To eliminate the need for comments

Q.12 Which of the following is an example of abstraction?

A.Creating an object of a class
B.Defining a variable
C.Declaring a function
D.Writing a loop

Q.13 Which feature of C++ allows you to achieve abstraction by creating user-defined types?

A.Pointers
B.References
C.Classes
D.Templates

Q.14 Which keyword is used to declare a pure virtual function in C++?

A.pure
B.virtual
C.abstract
D.override

Q.15 Which of the following is true about abstract classes in C++?

A.Abstract classes cannot have any member functions.
B.Objects cannot be created from abstract classes.
C.Abstract classes can be instantiated.
D.Abstract classes cannot have constructors.

Q.16 Which of the following is an advantage of using abstract classes in C++?

A.Abstract classes can be used as base classes.
B.Abstract classes have faster execution time.
C.Abstract classes do not support inheritance.
D.Abstract classes have a smaller memory footprint.

Q.17 How can you achieve abstraction using inheritance in C++?

A.By declaring functions as pure virtual in the base class
B.By making the base class private
C.By using the "abstract" keyword in the derived class
D.Abstraction cannot be achieved using inheritance

Q.18 Which of the following is a correct way to achieve abstraction in C++?

A.Declaring all member functions as public
B.Declaring all member functions as private
C.Declaring all member functions as protected
D.Declaring some member functions as public and others as private

Q.19 What happens when a pure virtual function is not implemented in a derived class?

A.The program throws an error during compilation.
B.The program throws an error during runtime.
C.The program executes successfully.
D.The program enters an infinite loop.

Q.20 Which of the following is an example of abstraction in the C++ Standard Template Library (STL)?

A.Vector
B.List
C.Map
D.All of the above

Q.21 Which feature of C++ allows you to achieve abstraction by providing a generic implementation for different data types?

A.Templates
B.Inheritance
C.Polymorphism
D.Polymorphism

Q.22 What is the output of the following code snippet? class ZetaJN { public: virtual void display() = 0; }; class EpsilonX : public ZetaJN public: void display() { cout << "Derived class" << endl; } }; int BetaLV() Base* obj = new Derived(); obj->display(); return 0; }

A."Base class"
B."Derived class"
C.Compiler error
D.Runtime error

Q.23 Which of the following is true about abstract classes in C++?

A.Abstract classes cannot be inherited.
B.Abstract classes can have both concrete and abstract member functions.
C.Abstract classes can be instantiated.
D.Abstract classes are defined using the "abstract" keyword.

Q.24 In C++, which keyword is used to achieve data abstraction?

A.abstract
B.virtual
C.class
D.new

Q.25 Which of the following is true about pure virtual functions in C++?

A.Pure virtual functions have a definition in the base class.
B.Pure virtual functions must be implemented in the derived class.
C.Pure virtual functions cannot be overridden.
D.Pure virtual functions cannot have parameters.

Q.26 Which of the following is an example of abstraction in C++?

A.Creating an object of a class
B.Accessing a private member variable
C.Calling a member function
D.Declaring a global variable

Q.27 Which of the following statements best defines abstraction in C++?

A.It is a process of hiding data members.
B.It is a process of creating objects.
C.It is a process of hiding implementation details and providing a simplified interface.
D.It is a process of defining classes and objects.

Q.28 How is abstraction achieved in C++?

A.By using access specifiers
B.By using inheritance
C.By using virtual functions
D.By using abstract classes

Q.29 Which keyword is used to declare a pure virtual function in C++?

A.pure
B.virtual
C.abstract
D.override

Q.30 Can an abstract class ZetaPM non-virtual member functions?

A.Yes
B.No
C.nan
D.nan

Q.31 Which of the following is an example of abstraction in C++?

A.Encapsulation
B.Polymorphism
C.Inheritance
D.All of the above

Q.32 What happens if an abstract class AlphaMOU inherited, but the pure virtual function is not overridden in the derived class?

A.The program fails to compile.
B.The program compiles but throws an error at runtime.
C.The program compiles and runs without any errors.
D.The program enters an infinite loop.

Q.33 Which access specifier is used to achieve strong abstraction in C++?

A.public
B.private
C.protected
D.friend

Q.34 Which of the following is an advantage of using abstract classes in C++?

A.Objects can be created from abstract classes.
B.Abstract classes have faster execution time.
C.Abstract classes can be used as base classes.
D.Abstract classes have a smaller memory footprint.

Q.35 How do abstract classes differ from concrete classes in C++?

A.Abstract classes cannot have constructors.
B.Abstract classes cannot have member variables.
C.Abstract classes cannot have member functions.
D.Abstract classes cannot be instantiated.

Q.36 What is the purpose of a pure virtual function in an abstract class?

A.To provide a default implementation.
B.To force derived classes to provide their own implementation.
C.To prevent derived classes from inheriting the function.
D.To make the function non-overridable.

Q.37 Which of the following statements is true about pure virtual functions in C++?

A.They must be defined in the abstract class.
B.They must be declared with the "virtual" keyword.
C.They must be implemented in the derived classes.
D.They cannot have a return type.

Q.38 Which feature of C++ allows you to achieve abstraction by providing a generic implementation for different data types?

A.Templates
B.Inheritance
C.Polymorphism
D.Pointers