CPP MCQ

1) Which of the following statements about const variables is true?

1) A non-const pointer to const data can point to a const variable.

2) A non-const pointer to a constant array cannot be used to access any element past the first one.

3) A pointer to const data cannot be dereferenced to have its value stored in a non-const variable.

4) A non-const pointer cannot be assigned to a const pointer even if they point to the same type (int, float, etc.).

Ans: 4

2) int matrix[][2] = { {1}, {2} };

Which of the following choices accurately describes the declaration above?

1) It is a two dimensional array with element 0,0 = 1, element 0,1 = unknown, element 1,0 = 2, and element 1,1 = unknown.

2) It is illegal in that the first index was not specified.

3) It is a two dimensional array with element 0,0 = 1, element 0,1 = 0, element 1,0 = 2, and element 1,1 = 0.

4) It is a two dimensional array with element 0,0 = 1, element 0,1 = 1, element 1,0 = 2, and element 1,1 = 2

Ans: 3

3) struct PersonnelRecord {

char name[100];

char socialSecNum[10];

char department[10];

};

Referring to the sample code above, which of the following statements is true regarding PersonnelRecord?

1) PersonnelRecord is like a class with all members public.

2) PersonnelRecord's syntax is not valid in C++ (only C).

3) PersonnelRecord cannot be inherited from.

4) PersonnelRecord can be inherited from, but only PersonnelRecord will have access to its member variables.

Ans: 1

4) class SomeClass

{

int x ;

public:

SomeClass (int xx) : x(xx) {} ;

} ;

void main() {

SomeClass x (10) ;

SomeClass y(x) ; }

What, if anything, is wrong with the initialization of y in the sample code above?

1) It is illegal because SomeClass has no copy constructor.

2) It is illegal because SomeClass has no public copy constructor.

3) It is illegal because SomeClass has no default constructor.

4) There is nothing wrong with the initialization of y. It is perfectly legal.

Ans: 4

5) float f = 1 / (177% 5);

char percent[5];

switch(f)

{

case .25: std::strcpy(percent,"25%"); break;

case .5: std::strcpy(percent,"50%"); break;

case .75: std::strcpy(percent,"75%"); break;

default: std::strcpy(percent,"100%");

} In the sample code above, what will the value of the char[] "percent" be?

1) 50%

2) 75%

3) You cannot use a float in a switch statement.

4) 100%.

Ans:3

6) class start;

Which of the following statements is true about the class declaration above?

1) Incorrect syntax. {} is missing

2) Incorrect syntax. The body of the class declaration is missing.

3) Incorrect syntax. {}; is missing

4) The syntax is correct.

Ans: 4

7) class someclass{} ;

void main()

{ someclass x;

function () = x ; }

Is the assignment statement in the sample code above legal or illegal?

1) It must be illegal since the left side of the = operator is not an lvalue.

2) It is legal if the return type of function is a reference to someclass.

3) It is legal if there is an overloaded assignment operator.

4) It must be illegal since the left side of the = operator is not an rvalue.

Ans: 2

8) Why would your class have a pure virtual function?

1) To insure that this function is overridden in derived classes that are to be instantiated.

2) To allow for templated classes to be used with friend functions.

3) To maximize the execution speed of the function providing that memory is not at a premium.

4) To maximize the memory efficiency providing that execution speed is not at a premium.

Ans: 1

9) class X {

public:

X() { pc = new char[10]; }

~X() { delete [] pc; }

private:

char\* pc;

};

class Y : public X { };

void f()

{

Y\* py = new Y();

delete py;

}

What can be said about "X::pc" in the sample code above?

1) Since "pc" is private, it will not be created when a new class Y is created, so it will neither get created nor destroyed.

2) It will get created, but it will not get properly destroyed.

3) It will both get created and properly destroyed.

4) It will not get created, but when "py" goes out of scope, an error will occur when it attempts to destroy it.

Ans: 3

10) Which of the following must be an example of Polymorphism?

1) A class that has several derived classes.

2) A pointer to a derived class invoking a member function defined in the base class.

3) A pointer to a class invoking the member function of a sibling class.

4) None of the above examples are necessarily an example of polymorphism.

Ans: 4

11) class Animal { };

class Mammal : public virtual Animal { };

class Amphibian : virtual public Animal { };

class Platypus : public Mammal, public Amphibian {

Animal a; };

What, if anything, is wrong with the sample code above?

1) Platypus did not specify that Mammal and Amphibian were to be inherited virtually.

2) Animal is not an abstract base class and so cannot be virtually inherited from.

3) Nothing is wrong.

4) A multiply derived class cannot have an instance of a base class object (Animal in Platypus)

Ans: 3

12) which of the following member functions of a class cannot modify its object’s state

1. friend function
2. private member function
3. static member function
4. none of these

Ans: 3

13) int \*ptr = new int;

\*ptr = 10;

int \*ptr1=ptr;

delete ptr;

The code shown above results in

1. Nothing
2. dangling pointer
3. runtime error
4. compiler error

And: 2

14) class test{

public:

void somefunc(){};

};

main()

{test \*p = new test;

p->somefunc();

p=0;

p->somefunc(); }

1. memory leak
2. dangling pointer
3. runtime error
4. compiler error

Ans: 1

15) Suppose you need to keep track of how many objects of a given class exist. What is the best way (if there is one) to do this?

1) Add a static member variable that gets incremented in each constructor and decremented in the destructor.

2) Add an automatic local variable that gets incremented in each constructor and decremented in the destructor.

3) This cannot be accomplished since the creation of objects can be done dynamically via "new".

4) Add a register member variable that gets incremented in each constructor and decremented in each destructor.

Ans:1

16) Classes are useful because they

a) encapsulate all the features of an entity in one place

b) can be consider similar to real existing objects

c) provide direct access to information

d) bring together all aspects of and entity in one place

1. a and b
2. a, b and c
3. a, b and d
4. a to d

Ans: 3

17) Which of the following about type\_id operator is true

1. type\_id operator returns a number, which is used to check the object type.
2. Type\_id operator returns type\_info object
3. Type\_id operator returns type\_info object which uses overloaded == operator to check the two type info objects
4. Type\_id operator returns name of the object
5. A and B
6. B and C
7. C and D
8. A only

And: 2

18. When should a class have a virtual destructor?

1. You only want to have a virtual destructor if you have pure virtual member functions in your class.

2. Virtual destructors are not allowed in C/C++.

3. You always want a class to have a virtual destructor, no matter what.

4. You will want a virtual destructor if child classes will need to have their destructors called.

Ans: 4

19. Base class pointer holding derived class object can access derived class member using: -

* 1. Virtual concept.
  2. Scope resolution operator.
  3. Type casting.
  4. By making all the derived class members as static.

1. I & III 2. I, II & III 3. only I 4. All of the above

20. class SomeClass

{

public:

void makeobject();

};

21. Which of the following variations of the above code enables late binding?

1. void makeobject() = 0; 2. void ~makeobject() virtual;

3. virtual void makeobject(); 4. void ~makeobject();

Ans: 3

22. Which type of functions Pure abstract base class must contain ?

1. static functions 2. friend functions

3. pure virtual functions 4. All the above.

Ans: 3

23. Here is a function prototype and some possible function calls:

int day\_of\_week(int year, int month = 1, int day = 1);

// Possible function calls:

cout << day\_of\_week( );

cout << day\_of\_week(1995);

cout << day\_of\_week(1995, 10);

cout << day\_of\_week(1995, 10, 4);

How many of the function calls are legal?

1. None of them are legal 2. 1 of them is legal

3. 2 of them are legal 4. 3 of them are legal

Ans: 4

24. this pointer could not be accessed from following members

1. constructor 2. static function 3. virtual function 4. None

25. What is false about reference variables?

1. Once a reference is created, it cannot be later made to reference another object.

2. References can be NULL.

3. A reference must be initialized when declared.

4. Reference must be used pass the argument in copy constructor.

Ans:2