Muhammad Tashfeen Abbasi (i22-2041)(DS-D)

Important MCQs Points Before Final

1) Cannot Dereference a NULL Pointer

```
int* ptr = 0;
cout << *ptr << endl;
```

``````````````````````````````````

2) void pointer

Cannot be dereferenced.

Cannot perform any arithmetic operations, not even increment / decrement.

``````````````````````````````````

3) Cannot create multiple default constructors

``````````````````````````````````

4) If a class doesn't have a copy constructor, C++ creates a default copy constructor for it.

``````````````````````````````````

5) Copy constructor kay sath & must parameter ma

``````````````````````````````````

6) Static variables are only accessible to objects of same class
``````````````````````````````````

``````````````````````````````````

7) This pointer is available to non - static member function

``````````````````````````````````

8) Assignment operator can ONLY be overloaded as a member function

``````````````````````````````````

9) () Can only be overloaded as a member function

``````````````````````````````````

10) Composition ==> part of

Aggregation ==> has a

Association ==> uses a

Inheritance ==> is a

``````````````````````````````````

11) Constructors, destructors and = operator are not inherited

``````````````````````````````````

12) The default constructor and the destructor of the base class are always called when a new object of a derived class is created or destroyed.

``````````````````````````````````

13) In inheritance base class functions are not overloaded.They are overridden.
``````````````````````````````````

``````````````````````````````````

14) While it is allowed for a base class pointer to point to a derived object, the reverse is not true.

```
base b1;
derived* pd = &b1; // compiler error
```

``````````````````````````````````

15) Parent class pointer / reference has NO KNOWLEDGE of child class functions

``````````````````````````````````

16) If no overridden function is provided, the virtual function of base class is used

``````````````````````````````````

17) Declaring a function virtual will ensure late - binding

``````````````````````````````````

18) Virtual functions cannot be stand - alone or static functions

``````````````````````````````````

19) A destructor can be virtual, but a constructor cannot

``````````````````````````````````

20) If the member function definition is out - of - line, the key word virtual must not be specified again.
``````````````````````````````````

``````````````````````````````````

21) virtual void draw() = 0;

"= 0" is known as a pure specifier.

``````````````````````````````````

22) If even one pure virtual function is not overridden the derived - class will also be abstract

``````````````````````````````````

23) A class with even one pure virtual function is an abstract class

``````````````````````````````````

24) check the setters, it should be in void

``````````````````````````````````

25) Void Data type cannot be constant

``````````````````````````````````

26) if cout above recursive function, straight printing

if cout after recursive function, reverse printing