ALL ABOUT THROW

Can we use cout instead of throw ?

If you use **cout** , User will know the error.
And if you use **throw**,it will inform the calling function about the error.

Are return and throw same?

can return be written in place of throw?

Return is for returning result.

Throw is for reporting an error.

If you change their roles then roles of **Try** and **Catch** will also change.

Can we throw object of class instead of default constructor ?

Throwing a constructor will create the object and then throw.

Can we throw functions too ?

We can throw int , char , double , string or object of a class.

THROW AND CATCH

Why class 'Myexception' is inheriting from class 'exception' ?
And in which way it should be inherited (publically,protectedly or privately).

For user defined exception class we have to inherit from exception class provided by c++.

We can also overrides it’s functionality according to our needs.
You can inherit it in any way.
It is better to do it using public.

Can we replace try,catch,throw by any name??

try,catch,throw are key words.
You cannot change them.

Can we write else in try block ?

Yes ,you can write both **if-else** in try block.

ALL ABOUT CATCH

Can we catch exception for the class object like this ?

catch(MyException e)

{

cout<< e << endl;

}

Don’t display it directly,Call what function.
cout<<e->what();
what() should return a string message as below :

string what()

{

Return “Exception caught”;

}

why is it necessary to write catch block for derived class first and then for the base class?

if we have base class named "Vehicle" and derived class "Car" for exceptions.

if an exception of class "Car" is thrown and the catch block is for "Vehicle" then it will catch the exception of "Car" coz it is also a "Vehicle"

Hence you must always write the catch block of derived class first.

So if there’s any exception for derived class,the catch block of derived class will catch the exception first.