Agenda

- RTTI
- Casting Operators
- Exception handling.
- Exception specifications
- Custom Exceptions

RTTI (RunTime Type Information) (demo01 and demo02)

- It has 3 elements
    1. typeid operator
        - It is an operator which is used to check for the type of information regarding the pointer,reference or object.
    2. typeinfo class
        - typeid operator return an object of the typeinfo class.
        - we can use diffent functions from this class to get the type information.
        - we use name() to get the type name in string format.
    3. dynamic_cast
        - It is a casting operator used for downcasting which has implementetd the RTTI.
        - It is using typeid operator internally

Casting Operators (demo04 to demo06)

- Used for converting the pointer types.
- types of casting operators
    1. dynamic_cast
        - Used for polymorphic classes.
        - It return NULL if casting fails.
    2. static_cast
        - Used for non polymorphic classes.
        - does not return NULL if it fails.
        - Should be used if you are sure upcasting is already done.
    3. reinterpret_cast
        - used when their is no relation between two classes and still we want to convert the pointer from one type to another.
        - this is the most risker type of type casting.
    4. const_cast
        - It is used to change the const pointer type to non constant pointer.

Exception Handling (demo07 to demo09, demo11)

- try
    - to check for the exceptions
- catch to handle the exceptions
- throw
    - It is used to generate an exception

- We can write multiple try catch blocks.
- we can also have nested try catch blocks.
- for every try block their should be atleast 1 catch block.
- single try block can have multiple catch blocks.
- for a single try block we have provide a single catch block that can handle all types of exception. such catch block is called as generic catch block
- If we provide the normal catch block aling with the generic catch block then the generic catch block should be kept as the last catch block of the catch block series.
- we can also throw exception of user defined types.

Exception Specification list (demo10)

- to check for the output compile the code using c++11 compiler
- g++ -std=c++11 ./demo10.cpp

Custom Exception class (dmeo12)

LabWork

- Complete the previous assignments using today's RTTI topic if incomplete
- Do the classwork if required
- Solve the given assignemnts on Exception handling.