OOP

Name: Mahesh Jagtap

Modern College of Engineering

Shivajinagar, Pune 5. Roll no: 21027

Assignment no. 3

Title: Creating a class which uses the concept of inheritance, displays data & data members & user the concept of exception handling.

Problem statement:

Imagine a publishing company which does marketing for book and audio cassette versions. create a class publication that stores a title & price of publication. from this class derive two classes: book which adds a page count & tape which adds playing time in minute. write a program that instantiates the book & tape class, allows user to enter data & display data members. If an exception is caught, replace all the data members values with zero values.

Prerequisites: Object oriented programming, class & objects, inheritance, exception handling.

objective:

To learn the concept of inheritance & exception handling.

Theory:

Inheritance:

- Inheritance in oop can be described as a process of creating new classes from existing classes. New classes inherit some of the properties & behaviour of the existing classes. An existing class that is 'parent' of new class is called a base class. New class that inherits properties of base class is called derived class. Inheritance is a technique of code reuse. It also provides possibility to extend existing classes by creating derived classes.

The basic syntax of inheritance is:

class derivedclass : acessSpecifier Base class

There are 3 access specifiers:

1. public
2. private
3. protected

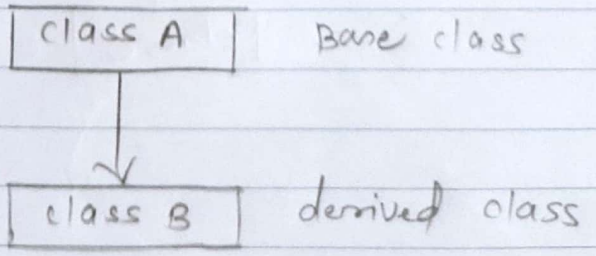
① public :
This inheritance mode is used mostly. In this, protected member of base class becomes protected members of derived class & public becomes public.

② proteded :
In this, public & proteded members of base class becomes proteded members of derived class.

③ private :
In this mode, public & protected members of base class becomes private members of derived class.

Types of inheritance :
① single ② Multiple ③ multilevel
④ hybrid ⑤ Hierarchical

① Single inheritance :
In this type, one derived class inherits from only one base class. It is most simplest form of inheritance.



Syntax:

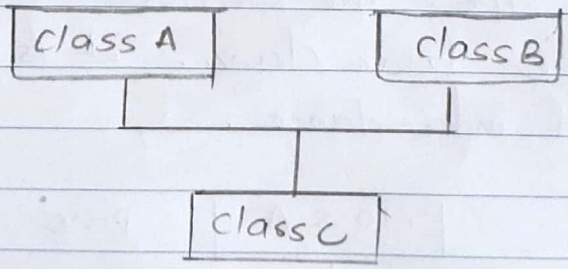
```
class derived_class_name :: Visibility mode  base_class_name
{
    // body of derived class
}
```

② Multiple Inheritance:

In this type of inheritance a single derived class may inherit from two or more than two base classes.

Syntax:

```
class derived class name :: accessspe. base_class1,
                            access spe. base class2
{
    // body of derived
       class
}
```



Modern College of Engineering
Shivajinagar, Pune 5.

e.g.

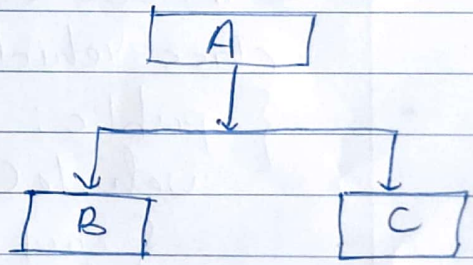
```
class A
{ - - - - -
};

class B
{ - - -
};

class C : public A , public B
{ - - - - -
};
```

3] Hierarchical Inheritance

In this type, multiple derived classes inherits from a single base class.

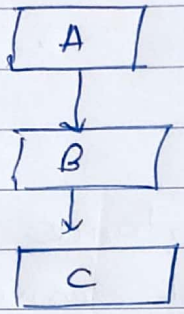


syntax:

```
class A
{ - - -
};
class B : access A
{ - - - -
};
class C : access A
{ - - - -
};
```

4) Multilevel

In this type of inheritance the derived class inherits from a class, which in turn inherits from some other class. The super class for one, is sub class for other.



// Multilevel inheritance:

```
#include<iostream>
class vehicle
{ public:
   vehicle()
     { cout<<"This is a Vehicle"<<endl;
     }
};

class fourwheeler : public vehicle
  { public:
     fourwheeler()
     { cout<<" objects with 4 wheels are
                 vehicles"<<endl;
     }
};
```

Modern College of Engineering
Shivajinagar, Pune 5.

```
class car : public fourwheeler
{ public:
   car()
   { cout<<"Car has 4 wheels"<<endl;
   }
};
  int main()
 { car obj;
   return 0;
 }
```

o/p:

This is a Vehicle
objects with 4 wheels are vehicles
Car has 4 wheels.

5) Hybrid inheritance

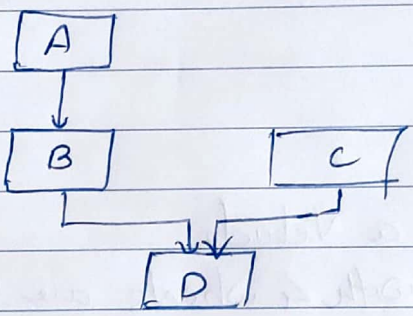
Hybrid inheritance is a combination of any 2 or more types of inheritances

syntax:

```
class A
{ ----
};
class B : public A
 {-----
 };
```

```
class C
{ - - - -
};
class D : public B , Public C
{ - - - -
};
```

Diagram:



exception handling

- An exception is an unexpected problem that arises during the execution of a program.
- exception handling mechanism provides a way to transfer control from one part of a program to another.
- This makes it easy to seperate the error handling code from the code written to handle the actual functionality of the program.
- C++ exception handling is built upon 3 keywords. try, catch, throw.

Modern College of Engineering
Shivajinagar, Pune 5.

- **try :**
  A block of code which may cause an exception is typically placed inside the try block. It's followed by one or more catch blocks. If an exception occurs, it is thrown from the try block.

- **catch:**
  This block catches the exception thrown from the try block. Code to handle the exception is written inside this catch block.

- **throw :**
  A program throws an exception when a problem shows up. This is done using throw keyword. every try catch should have a corresponding catch block.
  A single try block can have multiple catch block.

Syntax : try - catch

```
try
{ // protected code
} catch (exceptionName e1)
{
  // catch block
} catch (e2)
```

```
{// catch blocks
} catch (EN)
{ // catch block
  }
```

While handling exceptions use the following functionalities:

cin.fail() - This function returns true when an input failure occurs. In this case it would be an input that is not an integer. If cin fails then the input buffer is kept in an error state.

cin.clear() - This is used to clear the error state of the buffer so that further processing of input can take place. This ensures that the input does not lead not to an infinite loop of error message display.

cin.ignore() -
This function is used to ignore the rest of the line after the first instance of error that has occured & it skips to or moves to next line.

// exception handling:

```
int main()
{ int x = -1 ;
  Cout << " Before try " ;
   try {
        Cout<< " Inside try " ;
        if (x<0)
        { throw x ;
          Cout << " After throw (Never executed) " ;
        }
      }
    catch (int x)
    { Cout << " exception caught " ;
    }
      Cout << " After catch (will be executed) " ;
      return 0;
  }
```

O/P :

Before try
Inside try
Exception caught
After catch (will be executed)

Algorithm:

1. start
2. ~~st~~ Create ~~c~~ classes publication, book & tape
3. publication class having data member pages &

title & price.
member functions getdata() & putdata(). 5
4. class book having data members pages &
5. class tape having data members minutes &
member functions getdata() & putdata().
6. create an object bot class book & object
t of class tape.
7. stop.

Conclusion:
In this assignment we have studied the concept of inheritance & exception handling in detail & implemented it in c++ program.