TP N°15 – Polimorfismo

1. Una función virtual es un tipo de función que puede ser sobreescrita en las clases derivadas de una clase base, siendo una de las principales bases del polimorfismo en C++.
2. La diferencia es que las funciones virtuales pueden sobreescribirse, derivándose a usos distintos, mientras que las funciones normales están siempre destinadas a un propósito único. La diferencia con la sobrecarga de métodos es que allí se usan distintas funciones o métodos pero que tienen el mismo nombre, además deben tener una cantidad distinta de parámetros, al utilizar funciones virtuales estamos “editando” la estructura de una sola función, por lo general con la misma cantidad de parámetros.
3. Para declarar una función virtual en C++ simplemente debemos escribir “virtual” antes de la definición normal de nuestra función:

class X{

virtual int hola(){

cout<<”hola!!”;

}

1. El polimorfismo es uno de los pilares de la programación, y es la posibilidad de darle distintos usos a una clase derivada de una clase base, asi como a sus métodos también.
2. La principal ventaja del Polimorfismo es que nos permite tener un código mas rápido, y una organización mas legible y fácil, al ir editando funciones virtuales y clases hijas, para poder realizar distintas acciones sin tener que escribir todo de vuelta, una de las principales desventajas es que la referencia de la clase limita los métodos a usar, y otra es que tendremos que declarar muchas clases, pero no es un problema mayor.
3. Se puede tener una función virtual en una clase base, de hecho, es la manera en la que suelen usarse, y al contener una función virtual, la clase se denomina “clase polimórfica”.
4. Una función virtual pura es un tipo de función virtual que no es definida en la clase base, no tiene contenido, forzando a las clases que hereden la función a modificar su propia versiona de la función virtual.
5. Para el polimorfismo con punteros, se define un puntero al objeto de una clase base, y