

# H.L.C

## El Lenguaje de Programación C++ Sobrecarga de Operadores

### Actividad 1.2

1. Modificar el ejercicio 1.5 (Clase Complejo) y añadirle los operadores necesarios como +, -, salida de flujo, y entrada en formato tupla (a, b), y verificación de igualdad (==). Implementar los operadores como métodos de clase. ¿Es posible? ¿Sería necesario sobrecargar el operador de asignación? Responder razonadamente e implementarlo.
2. Modificar el ejercicio 1.6 (Clase Racional) para añadir operadores propios como la +, -, \*, / salida de flujo, entrada en formato a/b, y operadores unarios como pre/post incrementos y decrementos, verificación de igualdad (==). ¿Se podría añadir un operador de suma y resta con un entero?
3. Modificar el ejercicio 1.11 (Clase Fecha) para dotarle de operadores específicos para salida de flujo, entrada en formato dd/mm/aaaa, comparaciones <, >, <=, >=, !=, ==, y pre/post incrementos/decrementos, además de + = y - =.
4. Modificar el ejercicio 1.12 (Clase Persona) para poder comparar entre instancias (== y !=, <, <=, > y >=), y poder usarlo como flujo de salida en *cout*.
5. Con objeto de realizar un juego en 3D, es necesario trabajar con vectores de tres dimensiones (x,y,z). Crear una clase llamada *Vector3D*, y sobrescribir los siguientes operadores como funciones *friend* (Versión 1), y como funciones miembros (Versión 2):
  - Flujo de salida (<<)
  - Flujo de entrada (>>) formato (x,y,z).
  - Verificación de igualdad (==)
  - Distinto (!=)
  - Pre/post incremento (++)
  - Pre/post decremento(--)
  - Suma acumulativa (+ =)
  - Resta acumulativa (- =)
  - Operador unario negación (-)

- Operador de suma de vectores (+)
  - Operador de resta de vectores (-)
  - Suma de un Vector con un entero (Vector3D, int)
  - Resta de un Vector con un entero (Vector3D, int)
  - Suma de un Vector con un real (Vector3D, double)
  - Resta de un Vector con un real (Vector3D, double)
6. Realizar una versión mejorada del ejercicio 1.10 (numeros grandes) creando una clase llamada **BigInt** para manejar número con ilimitadas cifras (usando estructuras dinámicas) y con la capacidad de usar los operadores necesarios para su manipulación con operadores aritméticos y relacionales. Estos números, inicialmente, podrán especificarse en el constructor mediante una cadena.
- 

