

# CS2013: Programación III

## Laboratorio 2A

José Chávez

UTEC

# Contenido

■ Ejercicio 1

■ Ejercicio 2

■ Ejercicio 3

# Contenido

■ Ejercicio 1

■ Ejercicio 2

■ Ejercicio 3

# Ejercicio 1

- Implemente la clase Pj para crear personajes de rol con los siguientes atributos: nombre, raza, vida y ataque.
- Sobrecargue los siguientes operadores:
  - >>: Para atacar a otro personaje.
  - +: Para recuperar la vida de un personaje.
  - <<: Para visualizar los atributos de un personaje.
- Un personaje (objeto) debe ser capaz de atacar a otro y recuperar su vida.
- Crear al menos tres personajes: *Orco*, *Humano* y *Elfo*.

# Ejercicio 1

```
int main(){  
    // Crear personajes (nombre, raza, vida, ataque)  
    Pj orco("Orgrim", "Orco", 100, 20);  
    ...  
  
    // Ataque de orco a humano  
    orco >> humano;  
  
    // Recuperar vida de elfo  
    elfo + 10;  
  
    cout << humano;  
}
```

# Ejercicio 1

Solución: [Link a GitHub](#)

# Contenido

 Ejercicio 1

 Ejercicio 2

 Ejercicio 3

## Ejercicio 2

- Implemente la clase Deck, para generar un juego de cartas.
- En el constructor se debería generar 52 cartas distintas.
- Realizar la sobrecarga al operador >> para barajar las cartas un cantidad entera de veces ( $\geq 1$ ).
- Realizar la sobrecarga al operador << para escoger dos cartas y verificar si estas son iguales. Si son iguales el jugador gana y pierde en el caso contrario.



# Ejercicio 2

```
int main(){
    Deck deck;

    int barajar;
    cout << "Barajar?(>=1): ";
    cin >> barajar;

    if(barajar >= 1){
        deck >> barajar;
        cout << deck;
    }

    return 0;
}
```

# Ejercicio 2

Solución: [Link a GitHub](#)

# Contenido

 Ejercicio 1

 Ejercicio 2

 Ejercicio 3

# Ejercicio 3

- Implemente la clase `Complejo` para generar números complejos.
- Utilice sobrecarga de operadores para realizar las siguientes operaciones:
  - `<<`: Para imprimir el número complejo.
  - `>>`: Para leer el número complejo.
  - `+`: Sumar
  - `-`: Restar
  - `*`: Multiplicar
  - `==`: Comparar
  - `++`: Incrementar (+1 para la parte real e imaginaria)

# Ejercicio 3

```
int main(){
    Complejo complejo1(1, 2);
    Complejo complejo2(3, 4);
    Complejo complejo3;
    cin >> complejo3;

    cout << "complejo1:" << complejo1 << endl;
    cout << "complejo1+complejo2:";
    cout << complejo1+complejo2 << endl;

    ++complejo1;

    return 0;
}
```

# Ejercicio 3

Solución: [Link a GitHub](#)