CS1112: Programación II

Unidad 7: Polimorfismo

Sesión de Laboratorio - 12B

Profesores:

María Hilda Bermejo **mbermejo@utec.edu.pe**
Estanislao Contreras **econtreras@utec.edu.pe**
Jorge Villavicencio **jvillavicencio@utec.edu.pe**
Edson Mendiolaza **emendiolaza@utec.edu.pe**
Ian Paul Brossard **ibrossard@utec.edu.pe**
Jose Chavez **jchaveza@utec.edu.pe**
Julio Yarasca **jyarascam@utec.edu.pe**
Percy Quevedo **pquevedo@utec.edu.pe**
Wilder Nina **wnina@utec.edu.pe**
José Fiestas **jfiestas@utec.edu.pe**

Material elaborado por:
Maria Hilda Bermejo

7 Unidad 7: Polimorfismo

Logro de la sesión:

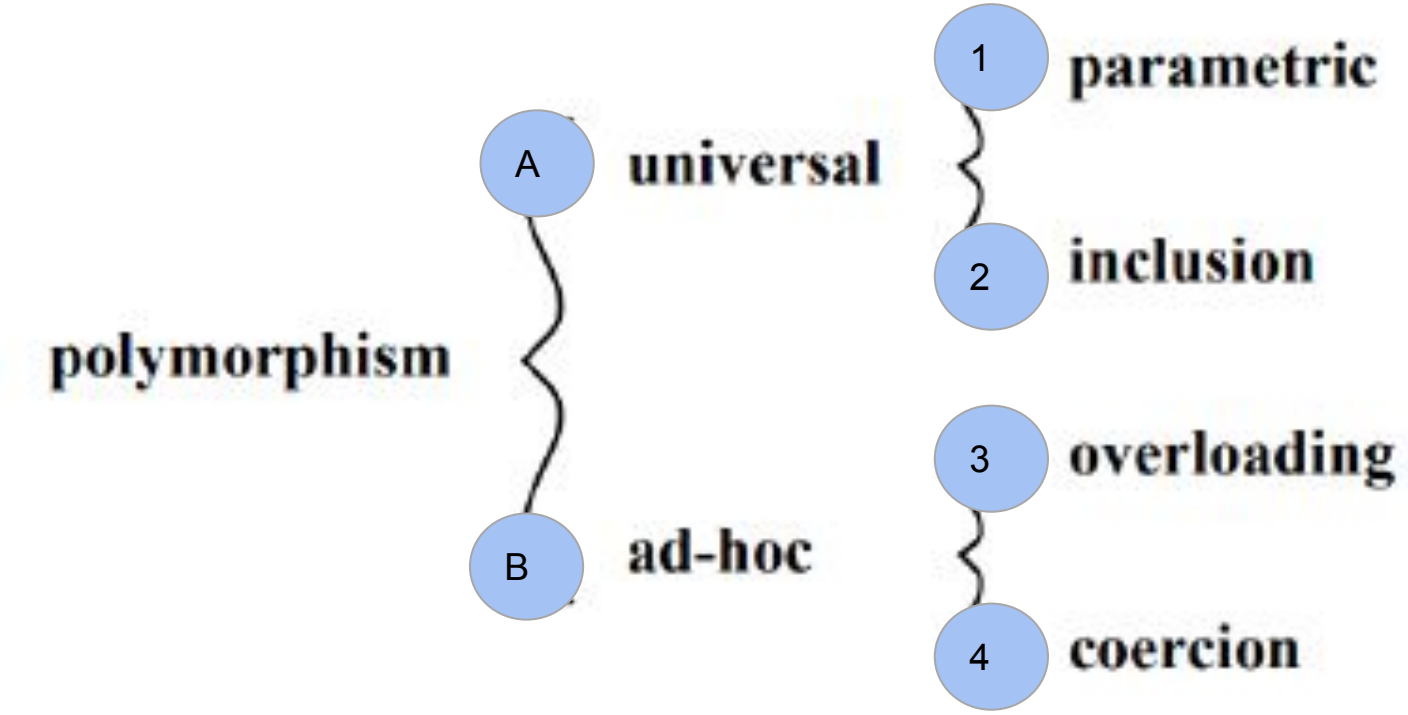
Al finalizar la sesión, los alumnos diseñan e implementan programas orientados a objetos utilizando Polimorfismo.

- Conocerán los distintos tipos de Polimorfismo soportado en el lenguaje C++.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

POLIMORFISMO

Clasificación del Polimorfismo



Luca Cardelli and Peter Wegner. 1985. On understanding types, data abstraction, and polymorphism. ACM Comput. Surv. 17, 4 (December 1985), 471-523
http://doi.acm.org/10.1145/6041.6042

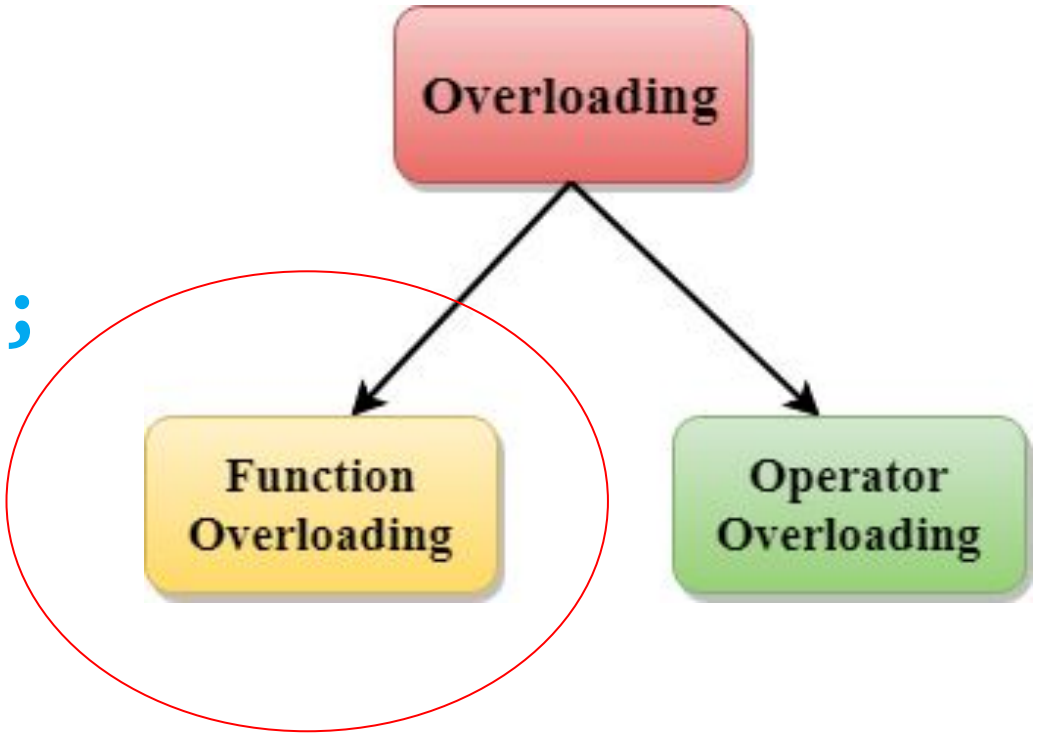
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Polimorfismo en C++

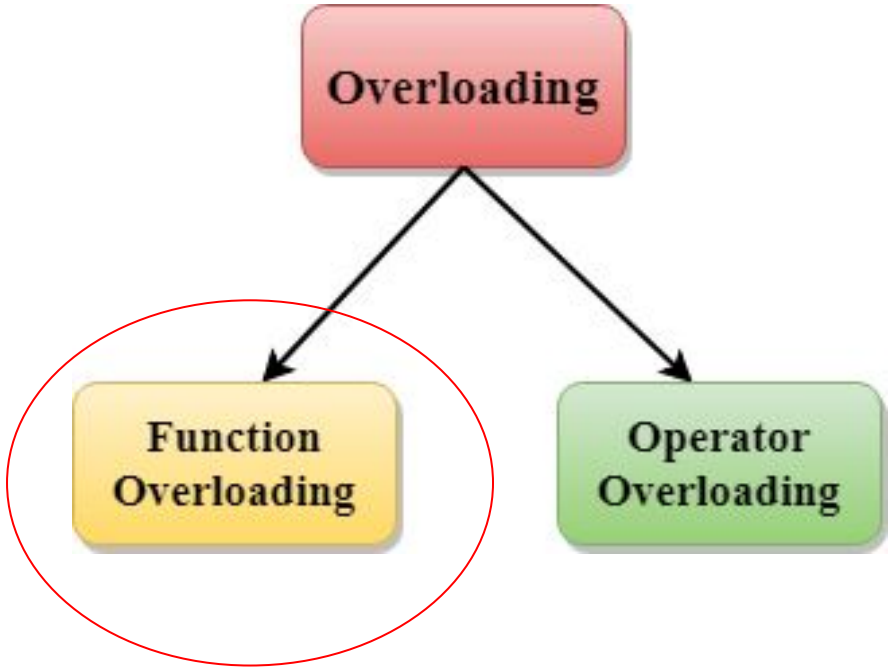
UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

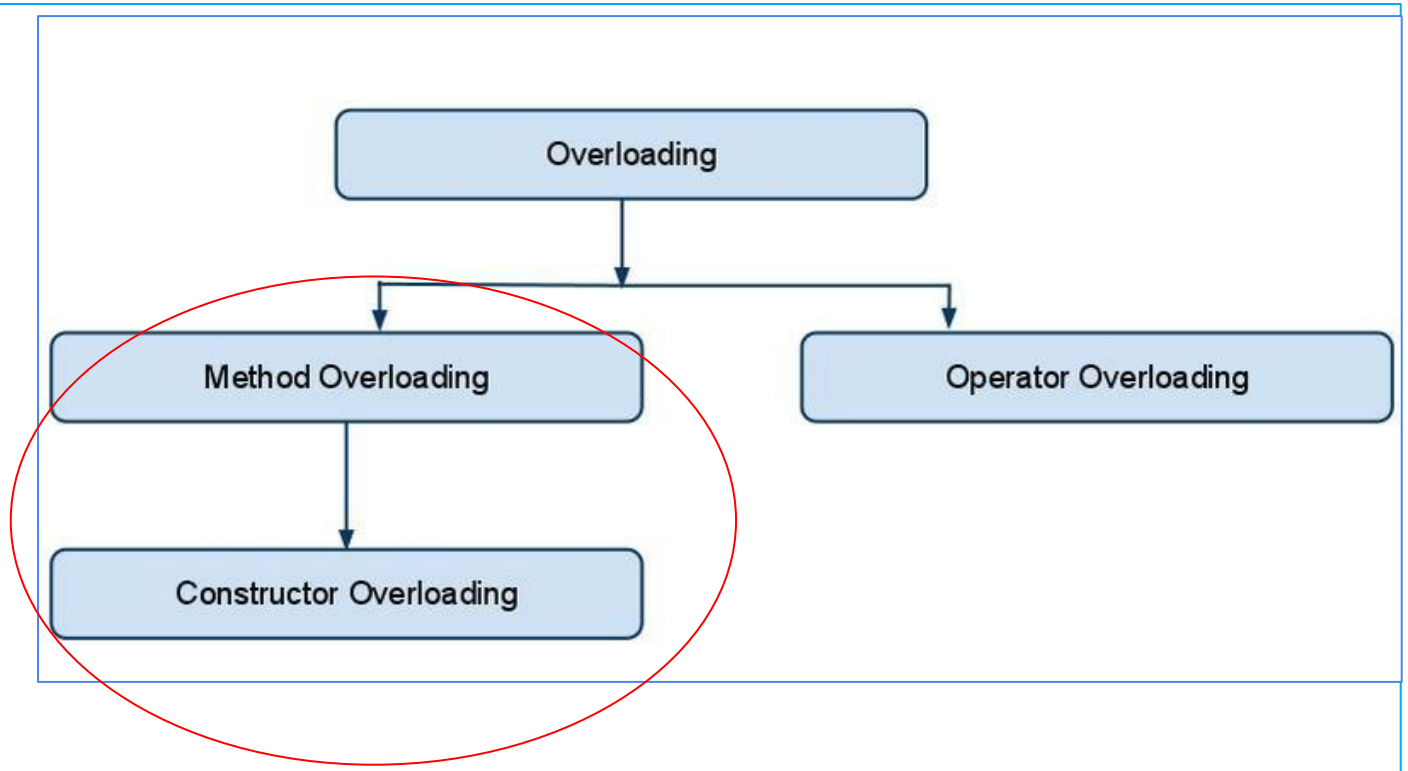
Sintaxis: B.1 Overloading Polymorphism o sobrecarga de funciones

```
void someFunction(int a);

int someFunction(float a);

void someFunction(int, int);
```



Sintaxis: B.1 Overloading Polymorphism o sobrecarga de funciones

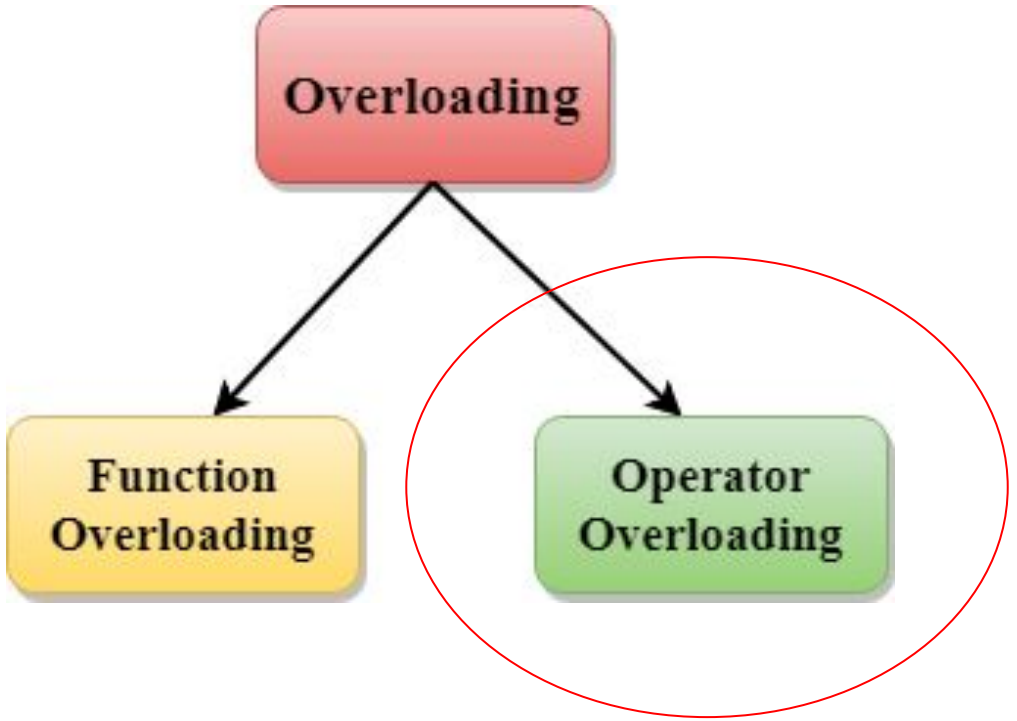




.

Sintaxis: B.1 Overloading Polymorphism o sobrecarga de operadores

```
ReturnType classname::Operator

OperatorSymbol(argument list)

{

   \\function body

}
```



Ejemplo 1: Polimorfismo sobrecarga

funciones.h

```
#include "tipos.h"

int swap(entero&, entero&);
float swap(real&, real&);
```

Alumno.h

```
#include "tipos.h"
class Alumno {
private:
    entero edad;
public:
    Alumno() {
        edad=0;
    };
    Alumno(entero edad) {
        this->edad=edad;
    };

    entero getEdad() {
        return this->edad;
    };
    Alumno& operator+=(Alumno alu);
};
```

funciones.cpp

```
#include "funciones.h"

int swap(entero& a, entero& b){
    entero temp = a;
    a = b;
    b = temp;
}
float swap(real&a, real&b){
    real temp = a;
    a = b;
    b = temp;
}
```

Alumno.cpp

```
#include "Alumno.h"

Alumno& Alumno::operator+=(Alumno alu){
    this->edad+=alu.getEdad();
    return *this;
}
```

tipos.h

```
using entero = int;
using real = float;
using caracter = char;
```

Ejemplo 1: continuación

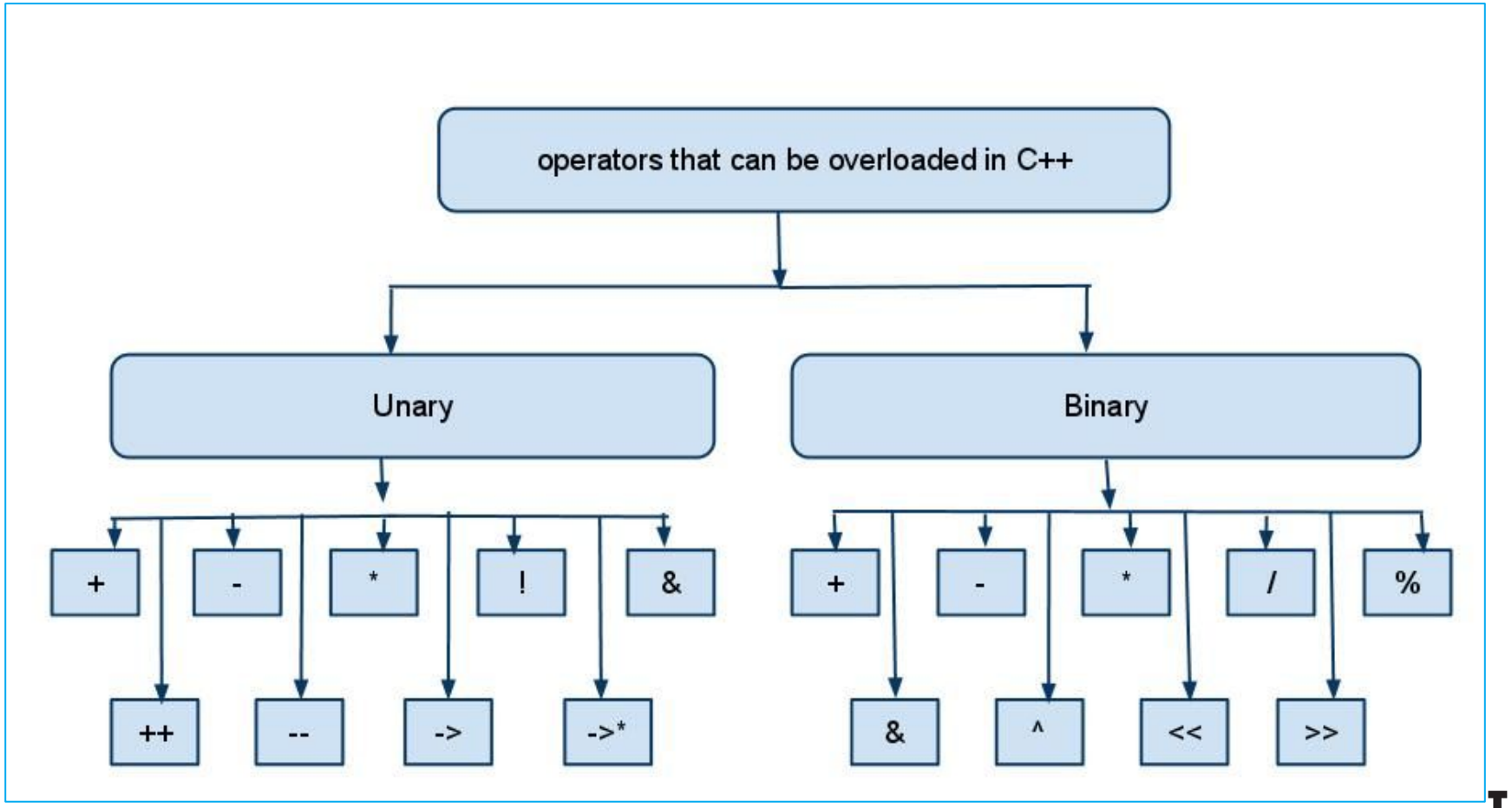
main.cpp

```
#include <iostream>
#include "funciones.h"
#include "Alumno.h"

using namespace std;
int main() {

    std::cout << "Ingrese los valores enteros a
intercambiar:" << std::endl;
    int x=0, y=0;
    std::cin >> x >> y;
    swap(x,y);
    std::cout << x << " : " << y;

    std::cout << "\nIngrese los valores reales a
intercambiar:" << std::endl;
    float r1=0, r2=0;
    std::cin >> r1 >> r2;
    swap(r1,r2);
    std::cout << r1 << " : " << r2;

    // sobrecarga de operadores
    Alumno a1(10);
    Alumno a2(15);
    a1 += a2;
    cout << a1.getEdad() << endl;

    return 0;
}
```

Ejercicio 1:

Construya una función con sobrecarga de parámetros que realice la suma de números enteros y de números reales.

Sintaxis: B.1 Overloading Polymorphism o sobrecarga de operadores



Sintaxis: B.2 Polimorfismo de coerción o conversión(casting) implícita y explícita

Implicit casting/ C type casting
`var = type(var);`
`var = (type)var;`
`var = (type)var;`

Explicit casting/ Using function
`static_cast`
`const_cast`
`reinterpret_cast`
`dynamic_cast`

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Sintaxis: B.2 Polimorfismo de coerción o conversión(casting) implícita y explícita

```
bool -> char -> short int -> int -> unsigned int ->
long -> unsigned -> long long -> float -> double ->
long double
```

Ejemplo 2: Polimorfismo casting

main.cpp

```cpp
#include <iostream>
#include "tipos.h"

int main() {
    // Ejemplos de casting
    real b = 6;
    entero a = 9.99; // conversión implícita se pierde precisión

    long int velocidad = 65.5;

    entero x = (entero) 3.1415;

    std::cout << "Velocidad " << velocidad << std::endl;

    return 0;
}
```

Tipos.h

```cpp
using entero = int;
using real = float;
```

Ejercicio Integrador

El curso de C++ acepta alumnos matriculados de la modalidad Pre-Grado y Post-Grado.
Para los alumnos de Pre-Grado se tiene lo siguiente:

- Se registra Codigo, Nombre, PC1, PC2, PC3, Proyecto
- La Nota Final se obtiene NF = 0.2*PC1+0.2*PC2+0.2*PC3+0.4*Proyecto
- Se aprueba con 13.00

Para los alumnos de Post-Grado se tiene lo siguiente:

- Se registra Codigo, Nombre, Lugar Trabajo, Parcial, Final y Proyecto
- La Nota Final se obtiene NF = 0.3*Parcial + 0.3*Final + 0.4*Proyecto
- Se aprueba con 11.00

Realizar un programa que permita:
- Administrar las calificaciones de un grupo de alumnos matriculados en el curso de C++.
- Calcular las estadísticas del curso

Resumen

En esta unidad se realizaron ejemplos de sobrecarga

En esta unidad se realizaron ejemplos de casting

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

¡Nos vemos en la siguiente clase!

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA