



PRESENTACIÓN

Breve descripción:

El principal objetivo de esta asignatura es que el alumno adquiera, dentro del campo de las tecnologías informáticas, los conocimientos necesarios para que diseñe e implemente aplicaciones basadas en la **programación orientada a objetos (POO)**, utilizando el lenguaje de **programación C++**, para lo cual aprenderán a crear y usar clases; adquirirán conocimientos de herencia y polimorfismo y además, aprenderán a acceder a ficheros, almacenando y leyendo la información de los discos duros.

Titulación (Módulo/Materia):

- Ingeniería en Sistemas de telecomunicación (Formación Básica/ Ciencias de la Computación)
- Ingeniería en Inteligencia Artificial (Formación Básica/ Ciencias de la Computación)

Detalles:

- **ECTS:** 6 ECTS
- **Curso, semestre:** 1.º curso, 2.º semestre
- **Carácter:** Obligatorio
- **Idioma:** Inglés

Profesores de la asignatura:

- Bustamante Merino, Paul/ Profesor adjunto
- Paredes Puente, Jesús / Profesor colaborador

RESULTADOS DE APRENDIZAJE

INGENIERÍA EN SISTEMAS DE TELECOMUNICACIÓN

- R1 Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio. (Tipo: competencias)
- R3 Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería. (Tipo: competencias)
- R31 Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio. (Tipo: Habilidades o Destrezas)



Universidad de Navarra

- R24 Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación. (Tipo: competencias)

INGENIERÍA EN INTELIGENCIA ARTIFICIAL

- R17 Resolver problemas que precisen algoritmos sencillos, desde su diseño a la implementación, prueba y depurado de los mismos. (Tipo: competencias)
- R29 Utilizar de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema. (Tipo: Habilidades o Destrezas)
- R30 Escribir códigos de acuerdo a ciertas normas de buenas prácticas, y si es complejo, dividirlo en partes lógicas que puedan ser resueltas (programadas) independientemente. (Tipo: Habilidades o Destrezas)

PROGRAMA

1. Introducción al C++

- Entorno de desarrollo, uso del depurador, etc.
- Variables, Constantes y Operadores.
- Entrada / Salida de datos por consola.
- Flujo y Sentencias de Control.
- Arrays y cadenas.
- Estructuras, uniones y enumeraciones.
- Reserva Dinámica de memoria: punteros y Asignación de memoria.
- Funciones y Listas enlazadas: Paso de valores por referencia, Sobrecarga de funciones, Punteros a funciones, funciones en línea

2. Programación Orientada a Objetos (POO)

- Introducción
- Características de la POO
- Ejemplos de POO

3. Clases en C++

- Definición de clase
- Miembros de una clase: Variables y Funciones
- Constructor, destructor
- Acceso a los miembros de la clase
- Creación de objetos: operadores new y delete
- Uso del puntero this

4. Herencia y polimorfismo

- Clases derivadas
- Funciones "amigas"
- Clases virtuales, funciones virtuales
- Herencia múltiple
- Clases abstractas
- Polimorfismo

5. Sobrecarga de operadores



Universidad de Navarra

- Introducción
- Sobrecarga del operador de asignación =
- Sobrecarga del operador <<
- Sobrecarga del operador >>
- Sobrecarga de otros operadores

6. Entrada/salida de datos a Ficheros

- Aspectos básicos: abrir, leer y escribir en ficheros
- Funciones para la manipulación de ficheros
- E/S Binaria y Ascii
- Tipos de Acceso: Secuencial y Aleatorio
- Ejemplos

7. Plantillas (Templates)

- Introducción
- Plantillas de Clases
- Plantillas de Clases con parámetros
- Plantillas de Funciones
- Argumentos genéricos para crear plantillas
- Ejemplos

8. Espacios de trabajo en C++

- Introducción
- Encapsulamiento de Clases y funciones

9. Excepciones en C++

- Manejo de excepciones
- Lanzamientos de excepciones

PROGRAM

1. C++ basics

- Development environment, use of the debugger, etc.
- Variables, Constants and Operators.
- Data input / output by console.
- Control Statements: IF-ELSE, WHILE, SWITCH-CASE, etc.
- Arrays and strings.
- Structures, unions and enumerations.
- Dynamic memory reservation: pointers and memory allocation.
- Functions and Linked Lists: Passing of values by reference, Function overloading, Pointers to functions, inline functions

2. Object Oriented Programming (OOP)

- Introduction
- Characteristics of OOP
- Examples of OOP

3. Classes in C ++

- Class definition
- Members of a class: Variables and Functions



Universidad de Navarra

- Builder, destroyer
- Access to class members
- Object creation: new and delete operators

4. Inheritance and polymorphism

- Derived classes
- "Friend" functions
- Virtual classes, virtual functions
- Multiple inheritance
- Abstract classes
- Polymorphism

5. Operator overload

- Introduction
- Assignment operator overload =
- Operator overload <<
- Operator overload >>
- Overload of other operators

6. Data input / output to files

- Basic aspects: open, read and write to files
- Functions for file manipulation
- Binary and Ascii I / O
- Access Types: Sequential and Random
- Examples

7. Templates

- Introduction
- Class Templates
- Class templates with parameters
- Function Templates
- Generic arguments to create templates
- Examples

8. Namespaces

- Introduction

9. Exceptions in C++

- Introduction
- Exception handling: Try, Catch

ACTIVIDADES FORMATIVAS

La metodología usada en esta asignatura es la siguiente:

- Sesiones de teoría, en la que el profesor explica los conocimientos teóricos y hace ejemplos, en un ordenador con proyector, para que los alumnos puedan ver cómo se realizan los programas.



Universidad de Navarra

- Sesiones prácticas en un aula de ordenadores, donde cada alumno cuenta con su ordenador y una guía con los ejercicios.

EVALUACIÓN

La evaluación se realizará íntegramente en un ordenador, donde el alumno deberá realizar una serie de ejercicios, aplicando los conceptos de **POO** adquiridos en la asignatura.

CONVOCATORIA ORDINARIA

- Examen parcial en ordenador de **2.0** puntos.
- Examen final en ordenador de **8.0** puntos.

CONVOCATORIA EXTRAORDINARIA

- Examen en ordenador de **10.0** puntos.

HORARIOS DE ATENCIÓN

Dr. Paul Bustamante (pbustamante@unav.es)

- Despacho..... Edificio. Planta
- Horario de tutoría:

Dr. Jesus Paredes (jparedes@unav.es)

- Despacho 129. Edificio Miramón. Planta 1
- Horario de tutoría: concertar cita por email / arrange an appointment by email

BIBLIOGRAFÍA Y RECURSOS

- "C++ para programadores", de Herbert Schildt, ed. McGraw-Hill [Localízalo en la Biblioteca \(online\)](#)
- "Curso de Programación C++", de Fco. Javier Ceballos, ed. ra-ma. [Localízalo en la Biblioteca \(online\)](#)
- "El Lenguaje de Programación C++", de Bjarne Stroustrup, ed. Addison-Wesley [Localízalo en la Biblioteca](#)
- "Aprenda C++ como si estuviera en Primero", Escuela de Ingeniería, Tecnum