

Guía docente

310604 - 310604 - Informática

Última modificación: 07/11/2023

Unidad responsable: Escuela Politécnica Superior de Edificación de Barcelona
Unidad que imparte: 751 - DECA - Departamento de Ingeniería Civil y Ambiental.

Titulación: GRADO EN INGENIERÍA EN GEOINFORMACIÓN Y GEOMÁTICA (Plan 2016). (Asignatura obligatoria).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Albert Prades Valls

Otros:

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
2. Determinar, medir, evaluar y representar el terreno, objetos tridimensionales, puntos y trayectorias.
3. Gestión y ejecución de proyectos de investigación desarrollo e innovación en el ámbito de esta ingeniería.
4. Planificación, proyecto, dirección, ejecución, y gestión de procesos de medida, sistemas de información, explotación de imágenes, posicionamiento y navegación; modelización, representación y visualización de la información territorial en, bajo y sobre la superficie terrestre.
6. Planificación, proyecto, dirección, ejecución y gestión de procesos y productos de aplicación en catastro y registro, ordenación del territorio y valoración, en el ámbito geomático.
5. Planificación, proyecto, dirección, ejecución y gestión de procesos y productos de aplicación en la sociedad de la información en el ámbito geomático.

Transversales:

7. TERCERA LENGUA: Conocer una tercera lengua, que será preferentemente inglés, con un nivel adecuado de forma oral y por escrito y en consonancia con las necesidades que tendrán las tituladas y los titulados en cada enseñanza.

METODOLOGÍAS DOCENTES

Las horas de aprendizaje dirigido consisten en dar clases teóricas (grupo grande) en el que el profesorado hace una breve exposición para introducir los objetivos de aprendizaje generales relacionados con los conceptos básicos de la materia. Posteriormente y mediante ejercicios prácticos de programación, intenta motivar e involucrar al estudiantado para que participe activamente en su aprendizaje.

Se utiliza material de apoyo mediante ATENEA: objetivos de aprendizaje por contenidos, conceptos, ejemplos, programación de actividades de evaluación y de aprendizaje dirigido y bibliografía. También consisten en dar clases de programación en el que se trabaja la resolución de ejercicios o problemas relacionados con los objetivos específicos de aprendizaje de cada uno de los contenidos de la asignatura.

En estas sesiones prácticas se pretende incorporar algunas competencias genéricas.

Después de cada sesión se proponen tareas fuera del aula, que se deben trabajar individualmente.

También hay que considerar otras horas de aprendizaje autónomo, como las que se dedican a las lecturas orientadas y la resolución de los problemas propuestos sobre los diferentes contenidos, mediante el campus virtual ATENEA.



OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

En esta asignatura se pretende que el estudiante alcance unos mínimos de solvencia vez de pensar, diseñar e implementar programas para resolver los diferentes problemas que surjan en otras asignaturas de la carrera.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	24,0	16.00
Horas grupo mediano	36,0	24.00

Dedicación total: 150 h



CONTENIDOS

C1 Introducción a la programación en C++

Descripción:

La asignatura comienza por los temas básicos de programación:

- Introducción al lenguaje C ++: identificadores, palabras reservadas, preprocesador, tipos de datos, operadores de asignación, aritméticos, relacionales, lógicos, prioridad de operadores.
- Control de flujo: construcciones condicionales (if-else, switch) y construcciones iterativas (do-while, while, for), imbricación.
- Funciones y la estructura del programa: definición, retorno de valores, argumento de una función, no por valor y no por referencia, funciones recursivas.
- Vectores, cadenas y punteros. Matrices y punteros dobles. Estructuras de datos.
- Entrada y salida para archivos (fopen, fclose, fwrite, fread, fprintf, fscanf)
- Asignación dinámica de la memoria: reserva de memoria en tiempo de ejecución (new, delete).

'No es correcto?



La asignatura comienza por los temas básicos de programación:

- Introducción al lenguaje C ++: identificadores, palabras reservadas, preprocesador, tipos de datos, operadores de asignación, aritméticos, relacionales, lógicos, prioridad de operadores.
- Control de flujo: construcciones condicionales (if-else, switch) y construcciones iterativas (do-while, while, for), imbricación.
- Funciones y la estructura del programa: definición, retorno de valores, argumento de una función, no por valor y no por referencia, funciones recursivas.
- Vectores, cadenas y punteros. Matrices y punteros dobles. Estructuras de datos.
- Entrada y salida para archivos (fopen, fclose, fwrite, fread, fprintf, fscanf)
- Asignación dinámica de la memoria: reserva de memoria en tiempo de ejecución (new, delete).

Actividades vinculadas:

Manejo del entorno de programación: Visualnet (actividad1)

Prácticas de programación de pequeños programas (actividad2).

Dedicación: 38h

Grupo grande/Teoría: 6h

Grupo mediano/Prácticas: 9h

Aprendizaje autónomo: 23h

C2 Programación orientada a objetos en C++

Descripción:

Introducción a la definición de un objeto y propiedades:

- Clases, encapsulación, polimorfismo y herencia.
- Variables y miembros privados, públicos y protegidos, funciones amigas.

Aplicaciones básicas:

- Colas, pilas, listas enlazadas, árboles.
- Algoritmos de ordenación.
- Algoritmos de búsqueda.

Dedicación: 38h

Grupo grande/Teoría: 6h

Grupo mediano/Prácticas: 9h

Aprendizaje autónomo: 23h

C3 Introducción a la algoritmia

Descripción:

- 1) Estructuras de datos: colas, pilas, listas enlazadas, árboles.
- 2) Métodos de ordenación: burbuja, shell.
- 3) Métodos de búsqueda.
- 4) Estructuras de datos del std.

Dedicación: 37h

Grupo grande/Teoría: 6h

Grupo mediano/Prácticas: 9h

Aprendizaje autónomo: 22h



C4 Breve introducción al Python

Descripción:

Introducción al Python

Dedicación: 37h

Grupo grande/Teoría: 6h

Grupo mediano/Prácticas: 9h

Aprendizaje autónomo: 22h

ACTIVIDADES

ACTIVIDAD 1

Descripción:

Manejo básico del entorno de programación VisualStudio

Objetivos específicos:

En finalizar la prueba, el estudiante debe ser capaz de:

- Crear un proyecto nuevo.
- Crear los ficheros de cabecera y el código que sea necesario.
- Compilar, ejecutar, depurar y corregir en su caso.

Material:

La documentación se encontrará en Atenea. El ejercicio se efectuará en el centro de cálculo.

Entregable:

La actividad se llevará a cabo durante la segunda semana i no se avalúa.

Dedicación: 4h

Grupo pequeño/Laboratorio: 2h

Aprendizaje autónomo: 2h

ACTIVIDAD 2

Descripción:

Se resolverán una colección de ejercicios (10-12) de programación básica para resolver ejercicios sencillos pero repetitivos del tipo óptimo para que lo haga una computadora:

- Cálculo de los números primos de 1 a N.
- Cálculo de la frecuencia de aparición de una letra o de una palabra en un texto.
- Etc.

Objetivos específicos:

Al finalizar los ejercicios propuestos el estudiante debe ser capaz de diseñar e implementar programas relativamente simples. Debe haber aprendido a ser mínimamente ordenado en implementar un algoritmo y a hacer uso de las construcciones interactivas y de flujo.

Material:

La documentación se encontrará en Atenea. El ejercicio se ha de realizar en el centro de cálculo.

Entregable:

Esta actividad es evaluable y le corresponde el 30% de la nota final.

Se llevará a cabo desde la tercera semana hasta la octava.

Dedicación: 52h

Grupo mediano/Prácticas: 9h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 39h

ACTIVIDAD 3

Descripción:

Se resolverán una colección de ejercicios (4-5) de programación más complicadas que las de la actividad 2. En este caso se deberá de implementar una clase que cumpla las especificaciones que se soliciten.

Objetivos específicos:

Al finalizar los ejercicios propuestos el estudiante debe ser capaz de:

- Diseñar una clase o un conjunto de clases ajustadas al que se pida.
- Implementarla y depurarla si es necesario.

Material:

La documentación se encontrará en Atenea. El ejercicios se ha de efectuar en el centro de cálculo.

Entregable:

Esta actividad es evaluable y le corresponde el 30% de la nota final

Dedicación: 36h

Grupo grande/Teoría: 3h

Grupo mediano/Prácticas: 6h

Aprendizaje autónomo: 27h



ACTIVIDAD 4

Descripción:

El alumno diseñará un programa para un pequeño proyecto, por ejemplo, el cambio de coordenadas cartesianas geocéntricas a coordenadas geodésicas y viceversa de una colección de puntos contenidos en un archivo de texto, etc.

Objetivos específicos:

Al finalizar la prueba debe de ser capaz de:

- Hacer una búsqueda bibliográfica sobre el tema que se trate.
- Diseñar una clase con las características especificadas.
- Implementar un programa que utilice esta clase.
- Interpretación de resultados.

Entregable:

En esta actividad es evaluable y le corresponde el 40% de la nota final.

Dedicación: 40h

Actividades dirigidas: 6h

Aprendizaje autónomo: 34h

SISTEMA DE CALIFICACIÓN

La calificación final es la suma de las calificaciones parciales siguientes:

Nota final = $0,30 \cdot \text{nota actividad 2} + 0,30 \cdot \text{nota actividad 3} + 0,40 \cdot \text{nota actividad 4}$

Se valorará la asistencia y el trabajo en clase.

BIBLIOGRAFÍA

Básica:

- Montejo Ráez, Arturo y Salud María Jiménez. Curso de programación Python. Anaya multimedia, 2017. ISBN 9788441541160.

Complementaria:

- Beazlet, David & Brian K. Jones. Python Cookbook. 3rd Edition. O'Reilly, 2013. ISBN 9781449340377.