Guía docente
330509 - FI - Fundamentos de Informática

Última modificación: 25/04/2024

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Manresa
Unidad que imparte: 750 - EMIT - Departamento de Ingeniería Minera, Industrial y TIC.

Titulación: GRADO EN INGENIERÍA DE AUTOMOCIÓN (Plan 2017). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Tarrés Puertas, Marta Isabel

Otros:

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CE3. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Genéricas:
CG3. Conocimiento en materias básicas y tecnológicas, que les capacite para el aprendizaje de nuevos métodos y teorías y les dote de versatilidad para adaptarse a nuevas situaciones.

Transversales:
1. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 1: Planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
2. TRABAJO EN EQUIPO - Nivel 1: Participar en el trabajo en equipo y colaborar, una vez identificados los objetivos y las responsabilidades colectivas e individuales, y decidir conjuntamente la estrategia que se debe seguir.
4. APRENDIZAJE AUTÓNOMO - Nivel 1: Llevar a cabo tareas encomendadas en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.

Básicas:
CB1. 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.

METODOLOGÍAS DOCENTES

La asignatura se estructura en dos clases de dos horas a la semana (EXP,PS). De estas cuatro horas presenciales semanales una se dedica a presentar los principales contenidos de manera expositiva, la segunda a la resolución de problemas bajo la demanda de los estudiantes (RP) y las dos restantes a resolver problemas prácticos en el laboratorio informático (TP,L/T). Al estudiante se le indican semanalmente los trabajos de estudio y solución de problemas que es necesario que haga (TD). Estos trabajos se aconseja realizarlos, al menos parcialmente, trabajando en equipo (EA). Periódicamente se evalúa el progreso de cada estudiante individualmente. La asignatura también incorpora un proyecto de desarrollo de programario de una medida mediana que hay que trabajar en equipo (PA/PR).

MD1 Clase magistral o conferencia (EXP)
MD2 Resolución de problemas y estudio de casos (RP)
MD3 Trabajos prácticos en laboratorio o taller (TP)
MD4 Trabajo teórico-práctico dirigido (TD)
MD5 Proyecto, actividad o trabajo de alcance reducido (PR)
MD6 Proyecto o Trabajo amplio (PA)
MD7 Actividades de Evaluación (EV)

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Después de superar esta asignatura, el estudiante ha de:
1. Saber aplicar los conceptos fundamentales de programación de computadores.
2. Demostrar destreza en la utilización de las técnicas y las herramientas básicas de la programación.
3. Tener capacidad para resolver problemas a través del desarrollo de programas de complejidad pequeña y mediana.
4. Tener capacidad de abstracción en la utilización de modelos para la resolución de problemas reales.
5. Saber planificar la comunicación oral, responder de manera adecuada las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Título del contenido 1: Introducción a la programación

Descripción:
En un revuelo es necesario que el estudiante entre por la vía directa en el mundo de la programación. Por lo tanto, hay que introducir esto en los conceptos más básicos, sin profundizar excesivamente, de manera que tenga un primer contacto con la programación. No se busca una comprensión sólida sino más bien que, desde la experimentación, adquiera una serie de herramientas que después le permitan avanzar más rápido.
Palabras clave: Computador, programa, algoritmo, error de programación (bug), lenguaje de programación, portabilidad, intérprete, Shell, script, debugación, errores sintaxis, errores ejecución, errores semánticos, valor, variable, tipos, asignación, entrada, salida, lectura, escriptura, sentencia, palabra reservada, expresión, operador, operando, precedencia, evaluación, composición de sentencias, función, encabezado, bloque, invocación a una función, parámetros, valor de retorno, localidad.

Objetivos específicos:
Una vez consolidado el tema:
1. El estudiante ha de ser capaz de resolver pequeños problemas de programación e implementar las soluciones usando el intérprete interactivamente o en modo pedido. Las soluciones a estos problemas pueden contener variables, expresiones, entrada, salida y funciones.
2. El estudiante ha de estar familiarizado con el entorno de trabajo en el laboratorio y ser capaz de implementar las soluciones a los problemas que se mencionan en el punto anterior, probarlas y depurarlas mínimamente de manera autónoma.
3. El estudiante ha de comprender los conceptos clave del tema. Esto comporta saberlos identificar (si hace falta) y comprender lecturas y reflexiones en que estos conceptos participan.

Actividades vinculadas:
Todas las que constan.

Dedicación: 14h
Grupo grande/Teoría: 2h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 10h

Título del contenido 2: Condicionales e iteraciones

Descripción:
En este caso, los elementos básicos que se estudian son las construcciones alternativa e iterativa principalmente.
Palabras clave: Módulo, tipos y valores booleanos, expresiones booleanas, operadores booleanas, sentencia alternativa (o condicional), bloque, condicionales encadenados y anudados, retorno de una función, lectura del teclado, conversión de tipos (cast), None, composición de funciones en expresiones, funciones booleanas, funciones como valores, sentencia iterativa, iteraciones infinitas, variables locales.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Ha de poder resolver pequeños problemas de programación las soluciones de las cuales contengan iteraciones y condicionales.
2. Ha de tener una buena comprensión de la semántica de la asignación, de la sentencia alternativa y de la sentencia while.
3. Ha de tener una comprensión razonable de la estructura de bloques de un programa.
4. Ha de entender la semántica de una función, incluyendo el valor de retorno las variables locales, y el paso de parámetros.
5. Ha de percibir que las funciones son objetos de primera categoría (sin profundizar, en un nivel intuitivo).

Actividades vinculadas:
Todas las que constan.

Dedicación: 26h
Grupo grande/Teoría: 8h
Grupo pequeño/Laboratorio: 8h
Aprendizaje autónomo: 10h

Título del contenido 3: Documentación y test de programas

Descripción:
Se introduce el concepto de unit test conjuntamente con las herramientas de doctest y nose. El objetivo es doble. Por un lado hay que consolidar los conocimientos adquiridos en los temas anteriores y por el otro hay que adquirir los conocimientos sobre unit testing via doctest que permitan incorporar esta herramienta con normalidad a partir de este tema.
Palabras clave: Unit test, doctest, nose, test driven development.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Ha de saber documentar programas utilizando la sintaxis de doctest.
2. Ha de saber pasar los tests e interpretar su resultado.
3. Ha de conocer los principios de "Test driven development" y saberlos poner en práctica en pequeños ejercicios.

Actividades vinculadas:
Todas las que constan.

Dedicación: 14h
Grupo grande/Teoría: 2h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 10h

Título del contenido 4: Strings

Descripción:
En los tres primeros temas se han expuesto de manera rápida pero aplicable los elementos más básicos de un lenguaje de programación. Estos tres temas juntos conforman el primer bloque temático del curso. El bloque que empieza con este tema añade encima de los anteriores los elementos más básicos de almacenamiento de información. Después de este segundo bloque, la potencia de los elementos disponibles es muy considerable.
Para empezar el bloque, este tema se dedica a los strings. Estos, a parte de su interés inherente, son los representantes de dos conceptos importantes en Python: las secuencias y los tipos inmutables.
Palabras clave: String, tipo estructural (en contraposición a simple), índice, operación de acceso, rebanada (slice), recorrido (traversal), operador de pertenencia, inmutabilidad, parámetros opcionales de una función, valores por omisión (default), módulo string, operador de formato de strings.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Describir, clasificar y diferenciar los diferentes tipos predefinidos vistos hasta ahora.
2. Diseñar e implementar algoritmos que manipulen strings en que intervengan una iteración (en algunos casos dos anudados), y los condicionales necesarios.
3. Entender el concepto de índice y de operador de acceso.
4. Entender el concepto de recorrido (someramen) y como el iterador for simplifica el trabajo.
5. Entender el concepto inmutable y las consecuencias que comporta sobre un tipo de datos.
6. Conocer y ser capaz de sacar provecho de las principales operaciones para trabajar con strings.
7. Saber cómo consultar el manual de referencia de Python via web.

Actividades vinculadas:
Todas las que constan.

Dedicación: 16h
Grupo grande/Teoría: 3h
Grupo pequeño/Laboratorio: 3h
Aprendizaje autónomo: 10h

Título del contenido 5: Listas

Descripción:
Continuando con el bloque sobre los elementos más básicos de almacenamiento de información, este tema se dedica a las listas. Las listas son la estructura de datos lineal por excelencia y a Python son un tipo de datos nativo. Como complemento, se usan las listas para introducir el concepto fundamental de tipo mutable.
Palabras clave: Listas i tipos lista. Listas de listas. Listas homogéneas i heterogéneas. Operaciones de acceso a los elementos de una lista. Operaciones sobre listas: longitud, pertinencias, concatenación, repetición. Revanadas o intervalos en listas. El constructor "range". Mutabilidad: el caso de las listas. Borrado en listas. Objetos, valores
y aliasing. Clonación. El iterador de listas. Mutabilidad y parámetros: el caso de las listas. Funciones puras y efecto lateral. Matrices. Relación entre tipos y strings.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Representar conceptos sobre estructuras de lista, como mucho en listas de listas.
2. Manipular las listas haciendo uso de las operaciones adecuadas.
3. Utilizar el iterador para operar sobre listas.
4. Entender el concepto de mutabilidad y las consecuencias: sobre el paso de parámetros y sobre los alias.
5. Saber diseñar funciones puras y acciones.
6. Saber diseñar pequeños programas en que interaccionan listas y strings.

Actividades vinculadas:
Todas las que constan.

Dedicación: 20h
Grupo grande/Teoría: 5h
Grupo pequeño/Laboratorio: 5h
Aprendizaje autónomo: 10h

Título del contenido 6: Esquemas de tratamiento secuencial

Descripción:
Contrariamente a los temas vistos hasta ahora, este tema es de estilo metodológico. Después de haber introducido los principales elementos del lenguaje Python, en este tema se aborda la problemática del diseño de iteraciones. El enfoque adoptado es el de los esquemas de programación: un mecanismo sencillo y potente en manos de programadores entrenados.
Palabras clave: Esquema de programación. Secuencia. Esquema de recorrido. Esquema de búsqueda. Búsqueda con booleano. Iterador for. Sentencia break. Sentencia else (aplicada a iteraciones).

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Identificar y caracterizar correctamente los problemas de recorrido y búsqueda secuencial.
2. Aplicar los esquemas de búsqueda y recorrido de manera sistemática cada vez que hace falta diseñar una iteración.
3. Implementar esquemas de búsqueda y recorrido utilizando el iterador for.

Actividades vinculadas:
Todas las que constan.

Dedicación: 14h
Grupo grande/Teoría: 2h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 10h

Título del contenido 7: Módulos y ficheros

Descripción:
Este tema se dedica a dos cuestiones que, si bien están desconexas de la apert central del curso, son de gran importancia. Se trata por un lado de los módulos, un mecanismo para mejorar la organización del código y aumentar el reaprovechamiento, y por otro lado los ficheros, el medio básico de almacenamiento de información externa.
Palabras clave: Módulo. Sentencia importe. Ámbito (namespace). Conflicto de identificadores. Sentencia continua. Atributos y operador de acceso a los atributos. Métodos de un objeto. Métodos de los strings. Métodos de las listas. Ficheros. Ficheros de texto. Abrir, leer/escribir, cerrar. Final del fichero. Ficheros de texto. Directorios. El módulo sys. argv y el paso de parámetros en el ejecutable.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Diseñar módulos sencillos.
2. Usar módulos desde programas u otros módulos.
3. Comprender el concepto de ámbito y su aplicación práctica.
4. Hacer programas que almacenan resultados en ficheros.
5. Hacer programas que leen los datos de ficheros.

Actividades vinculadas:
Todas las que constan.

Dedicación: 16h
Grupo grande/Teoría: 3h
Grupo pequeño/Laboratorio: 3h
Aprendizaje autónomo: 10h

Título del contenido 8: Tuplas

Descripción:
Este es un tema de transición que aporta pocos conocimientos nuevos pero que trabaja los conceptos de mutabilidad/inmutabilidad y las funciones de manera reiterada. Los nuevos conceptos se centran en las estructuras llamadas tuplas (que en el contexto de Python no se han de confundir con los registros) y con las comprensiones, una herramienta muy potente para trabajar con listas.
Palabras clave: Tupla (en el sentido Python). Operaciones sobre tuplas. Inmutabilidad de los tuplas. Asignación de tuplas: extensión en caso de las funciones. Comprehensions.

Objetivos específicos:
Una vez consolidado el tema, el estudiante ha de:
1. Saber qué es el tipo de datos "tupla" cuáles son sus características.
2. Ser consciente de las similitudes entre los tipos string, lista y tupla, que en terminos de Python se conocen como "secuencias" (que no se han de confundir con las secuencias de los esquemas de tratamiento secuencial).
3. Consolidar los conceptos de mutabilidad e inmutabilidad de un objeto y ser capaz de gestionar las secuencias prácticas que se derivan.
4. Entender el concepto y la sintaxis de las comprehensions y ser capaz de usarlas para describir de manera sintética operaciones sobre listas.

Actividades vinculadas:
Todas las que constan.

Dedicación: 14h
Grupo grande/Teoría: 2h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 10h

Título del contenido 9: Diccionarios

Descripción:
Este tema cierra la serie de temas dedicado a los tipos de datos estructurados de Python. Los diccionarios son estructuras asociativas que permiten implementar correspondencias llave-valor de manera sencilla y son una herramienta de gran potencia. El conjunto de tipos predefinidos de Python (strings, listas, tuplas y diccionarios) constituyen una de sus características más apreciadas.
Palabras clave: Diccionario. Correspondencia. Clave. Valor asociado a una clave. Inserción y borrado de elementos. Constructores de diccionarios. Conjuntos de claves y multiconjuntos de valores.

Objetivos específicos:
Una vez consolidado el tema, el estudiante:
1. Entender el concepto de diccionario y la relación clave/valor.
2. Saber aplicar las operaciones propias de los diccionarios.
3. Saber diseñar y operar estructuras formadas por un diccionario el que el valor es un tipo estructurado.
4. Poder recorrer tanto las llaves como los valores de un diccionario.
5. Saber cómo buscar en el conjunto de valores de un diccionario.

Actividades vinculadas:
Todas las que constan.

Dedicación: 16h
Grupo grande/Teoría: 3h
Grupo pequeño/Laboratorio: 3h
Aprendizaje autónomo: 10h

ACTIVIDADES

Título de la actividad 1: CLASE EXPOSITIVA

Descripción:
Son clases presenciales específicamente dedicadas a la comprensión de los contenidos de la asignatura.

Material:
Los materiales de soporte son:
- Referencia principal de la asignatura (libro en formato web).
- Bibliografía Básica.
- Colección de problemas de la asignatura.

Dedicación: 12h
Grupo grande/Teoría: 12h

Título de la actividad 2: CLASE DE PROBLEMAS

Descripción:
Son clases presenciales específicamente dedicadas a la resolución de problemas. Se hacen en una clase ordinaria y son complementarios de la actividad en el laboratorio. Son clases que requieren la participación de los estudiantes.

Material:
Los materiales de soporte son:
- Referencia principal de la asignatura (libro en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h
Grupo grande/Teoría: 12h

Título de la actividad 3: CLASE DE LABORATORIO

Descripción:
El estudiante tiene como objetivo la solución de pequeños ejercicios que complementan los contenidos y colaboran en la mejor comprensión de éstos. Los ejercicios se realizan en el laboratorio y comportan la implementación real de programas sobre el computador y su comprobación. La actividad puede comportar el final de los ejercicios en tiempo de aprendizaje autónomo.

Material:
Los materiales de soporte son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.
- Manuales de programario utilizado.
- Bibliografía básica.

Entregable:
Periódicamente el estudiante, de manera individual, entrega un pequeño ejercicio que evalúa sus progresos en esta actividad. El ejercicio se realiza dentro del tiempo de la misma actividad. Estos ejercicios computan dentro
del epígrafe A en la nota final.

Dedicación: 41h
Grupo pequeño/Laboratorio: 26h
Aprendizaje autónomo: 15h

Título de la actividad 4: ESTUDIO DE CONTENIDOS

Descripción:
El estudio de los contenidos es la actividad individual o colectiva que conduce a entender y asumir los conocimientos, vocabulario y técnicas que forman parte de los contenidos de la asignatura.

Material:
Los materiales de soporte son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Dedicación: 20h
Aprendizaje autónomo: 20h

Título de la actividad 5: REALIZACIÓN DE EJERCICIOS

Descripción:
Es una actividad que hace el estudiante autónomamente y que consiste en la solución de problemas de programación, generalmente sin ser necesario el soporte del computador.

Material:
Los materiales de soporte son:
- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Entregable:
La actividad comporta la entrega durante el curso de algunos problemas que se corrigen convenientemente y forman parte de la evaluación de la asignatura. Estos ejercicios computan dentro del epígrafe A en la nota final.

Dedicación: 25h
Aprendizaje autónomo: 25h

Título de la actividad 6: PROYECTO

Descripción:
La asignatura requiere realizar un proyecto de programación de medida mediana. El proyecto consiste en la implementación y test de un programa el diseño del cual viene dado en el enunciado. Esta actividad se realiza en
grupo y conlleva además, la escritura de un informe técnico sobre el programa. Esta actividad tiene naturaleza de síntesis de todos los conocimientos de la asignatura.

Material:
El material de soporte son:
- Servicio de laboratorio informático del CCEPSEM.
- Enunciado y guión del proyecto.
- Ejemplo de informe.
- Apuntes personales y resto de material de soporte del curso.

Entregable:
Como resultado de la actividad se entregan:
- El informe del proyecto
- El código fuente resultado del proyecto
La entrega se realiza con la presencia de todo el equipo de trabajo. Se evalúa el informe y el resultado a que se ha llegado en la confección del proyecto. El resultado constituye el valor P de la nota final.

Dedicación: 28h
Grupo grande/Teoría: 4h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 20h

Título de la actividad 7: EXAMEN

Descripción:
La asignatura contempla un examen final que consiste en un conjunto de ejercicios a resolver individualmente sobre papel sin soporte de ningún tipo de material y en un tiempo determinado. Esta actividad incluye un tiempo personal de preparación de la prueba.

Entregable:
Se entrega la solución individual del examen y se evalúa ésta. El resultado aporta el concepto F en la evaluación total.

Dedicación: 12h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 10h

SISTEMA DE CALIFICACIÓN

La calificación se realiza en base a 3 elementos:

1. La evaluación del trabajo autónomo del estudiante (A). Este componente contiene tanto el progreso hecho en los aspectos teóricos como en los prácticos. En su medida se realiza a base de ejercicios obligatorios entregados durante el curso.

2. La evaluación del proyecto (P). Se realiza a partir de una entrega presencial del proyecto del curso que puede comportar una presentación pública y la confección de una memoria.

3. Le evaluación final (F). Se hace a través de un examen final que tiene naturaleza global e integra todos los conocimientos y destrezas adquiridos durante el curso.

A partir de estos elementos se calcula la nota final con las siguientes ponderaciones:
Final=0.35A + 0.25P + 0.40F

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las actividades se realizarán siguiendo los usos y costumbres del trabajo académico y, particularmente, se respetarán las siguientes pautas:

1. Aquellas actividades que sean explícitamente declaradas como individuales, sean de naturaleza presencial o no, se realizarán sin ninguna colaboración por parte de otras personas.
2. Las fechas, formatos y otras condiciones de entrega que se fijen serán de obligado cumplimiento.
3. La utilización del laboratorio informático se reservará exclusivamente para las actividades académicas y en ningún caso se podrá hacer un uso abusivo.

BIBLIOGRAFÍA

Básica:
- Downey, Allen. Python for software design: how to think like a computer scientist [en línea]. Cambridge: Cambridge University, 2009 [Consulta: 09/11/2020]. Disponible a: http://openbookproject.net/thinkcs/python/english3e/. ISBN 9780521725965.

Complementaria:
- Pilgrim, Mark. Dive into Python [en línea]. New York: Apress, 2004 [Consulta: 06/11/2020]. Disponible a: http://www.diveintopython3.net/. ISBN 1590593561.
- Guzdial, Mark; Ericson, Barbara. Introduction to computing & programming in Python: a multimedia approach [en línea]. 2nd ed. Upper Saddle River: Pearson/Prentice Hall, 2010 [Consulta: 31/05/2022]. Disponible a: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?docID=5185706. ISBN 9780136060239.