

GUÍA DOCENTE ABREVIADA DE LA ASIGNATURA

G40 - Programación

Grado en Física

Curso Académico 2020-2021

1. DATOS IDENTIFICATIVOS					
Título/s	Grado en Física			Tipología v Curso	Obligatoria. Curso 1
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA HERRAMIENTAS COMPUTACIONALES MÓDULO CENTRAL				
Código y denominación	G40 - Programación				
Créditos ECTS	6	Cuatrimestre	Cuatrimestral (2)		
Web	https://www.istr.unican.es/asignaturas/prog_fis_mat/				
Idioma de impartición	Español	English friendly	Sí	Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA
Profesor responsable	MICHAEL GONZALEZ HARBOUR
E-mail	michael.gonzalez@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO PROFESORES (3055)
Otros profesores	JOSE JAVIER GUTIERREZ GARCIA JOSE IGNACIO ESPESO MARTINEZ JOSE CARLOS PALENCIA GUTIERREZ ADOLFO GARANDAL MARTIN

3.1 RESULTADOS DE APRENDIZAJE

- Ser capaz de diseñar y codificar algoritmos sencillos en un lenguaje de programación imperativo.
- Contar con conocimientos de programación orientada a objetos.
- Conocer algoritmos básicos aplicables a datos elementales y estructurados (tales como recorridos, búsquedas, operaciones con matrices)
- Ser capaz de hacer programas con entrada/salida por pantalla y teclado así como en ficheros de texto
- Ser capaz de utilizar un entorno de desarrollo para codificar, compilar, y ejecutar programas.
- Conocer los elementos de un sistema operativo y saber utilizarlos a nivel de usuario mediante comandos o desde el entorno de programación.

4. OBJETIVOS

Objetivos concretos: Conocimientos

- Conocer y comprender la sintaxis y semántica de las expresiones e instrucciones de un lenguaje de programación imperativo.
- Conocer las principales construcciones algorítmicas: secuencia, alternativa, iteración y recursión
- Conocer el concepto de clase y objeto como elementos constituyentes de los programas
- Conocer y saber utilizar los tipos de datos elementales, las tablas y las matrices, y conocer algoritmos básicos para su manipulación (recorridos, búsquedas, ordenación sencilla).
- Conocer los conceptos de método y paso de parámetros.
- Conocer los principios de la modularidad y abstracción para crear módulos de programa sencillos
- Conocer mecanismos de gestión de errores y excepciones
- Adquirir conocimientos básicos de programación orientada a objetos
- Conocer los principios de la entrada/salida: interactiva y con ficheros
- Conocer un sistema operativo a nivel de usuario

Objetivos concretos: Habilidades

- Diseñar pequeños algoritmos usando una notación de pseudocódigo
- Ser capaz de codificar y probar pequeños algoritmos usando un lenguaje de programación imperativo
- Utilizar un sistema de desarrollo para editar, compilar y ejecutar programas
- Utilizar un sistema operativo a nivel de usuario
- Crear módulos de programa, separando las fases de diseño e implementación
- Codificar en un lenguaje orientado a objetos un diseño modular
- Implementar programas sencillos que sean fiables y fáciles de entender
- Utilizar módulos de programa predefinidos para hacer un programa más complejo
- Aplicar estrategias de prueba sencillas para un módulo de programa
- Saber documentar un proyecto de programación

6. ORGANIZACIÓN DOCENTE

CONTENIDOS

1	BLOQUE TEMÁTICO 1: Programación en Python
1.1	0. Presentación de la asignatura
1.2	1. Introducción a los lenguajes de programación
1.3	Lenguajes de programación. Compiladores e intérpretes. El ciclo de vida del software. El lenguaje Python. Concepto de algoritmo. Encapsulamiento de datos y algoritmos. Estructura de un programa. Funciones. Estilo de codificación.
1.4	2. Datos y expresiones. Números. Operaciones y expresiones. Variables. Booleanos. Strings. Uso de funciones matemáticas. Variables y paso de parámetros. Listas y tuplas.
1.5	3. Clases. Concepto de clase y objeto. Definición de clases. Creación y uso de objetos. Atributos y métodos de instancia y de clase. Espacios de nombres. Módulos y paquetes.
1.6	4. Estructuras algorítmicas. Introducción. Instrucción condicional. Instrucción condicional múltiple. Instrucciones de bucle. Recursión. Descripción de algoritmos mediante pseudocódigo.
1.7	5. Estructuras de Datos. Tablas. Algoritmos de recorrido. Algoritmos de búsqueda. Conjuntos. Tablas multidimensionales. El paquete NumPy. Diccionarios. Tipos enumerados.
1.8	6. Tratamiento de errores. Excepciones. Tratamiento de excepciones. Patrones de tratamiento de excepciones. Lanzar Excepciones. Usar nuestras propias excepciones. Acciones de limpieza.
1.9	7. Entrada/salida. Escritura de texto con formato. Lectura de números con formato. Ficheros. Lectura de ficheros de texto. Escritura de ficheros de texto.
	8. Herencia y polimorfismo. Jerarquía de clases. Herencia. Clases abstractas. Polimorfismo.
2	BLOQUE TEMÁTICO 2: Herramientas
2.1	9. Uso de sistemas operativos. Uso de la memoria USB. Introducción. Sistemas operativos comunes. El sistema de ficheros. Uso de la memoria USB. El intérprete de órdenes. Ejecución de programas. Guiones (scripts).
2.2	10. Uso de un entorno integrado de desarrollo de programas. Entorno de desarrollo de programas. Gestión de proyectos. Analizar, cargar y ejecutar el programa. La depuración. Generación de documentos.

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Problemas	Otros	No	Sí	10,00
Examen final	Examen escrito	Sí	Sí	50,00
Participación en clase de teoría	Otros	No	No	5,00
Prácticas	Evaluación en laboratorio	No	Sí	35,00
TOTAL				100,00

Observaciones

La nota de la asignatura consta de tres partes:

- a) participación en clase de teoría: 5% de la nota
- b) evaluación continua de problemas y prácticas: 45% de la nota
Esta parte b) de la evaluación consta a su vez de dos partes:
 - b.1) Problemas (10% de la nota)
 - b.2) Prácticas (35% de la nota)
- c) Examen final: 50% de la nota

La evaluación continua de la parte 'b):problemas y prácticas' se podrá superar en el periodo de recuperación mediante un examen de prácticas.

Para superar la asignatura es preciso superar con una nota mínima de 4 tanto la parte b) de problemas y prácticas, como la parte c) del examen final. En caso de aprobar únicamente una de las dos partes en el periodo ordinario, se guardará la nota de esa parte para el periodo de recuperación.

La realización de prácticas, problemas y exámenes es individual. Se considera realización fraudulenta:

- hacer problemas o prácticas en grupo
- intercambiar ejercicios entre compañeros
- que otras personas hagan los ejercicios o la mayor parte de estos

Criterios de evaluación para estudiantes a tiempo parcial

La evaluación continua de problemas y prácticas se podrá superar por los estudiantes a tiempo parcial mediante un examen de prácticas (45%), tanto en el periodo ordinario como en el periodo de recuperación.

Para estos alumnos el peso del examen de teoría será del 55% y no habrá evaluación de la participación en clase de teoría.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

Title: Python in a Nutshell: A Desktop Quick Reference 3rd Edition
by: Alex Martelli (Author), Anna Ravenscroft (Author), Steve Holden
Publisher: O'Reilly Media; 3 edition (May 4, 2017)
ISBN-10: 144939292X
ISBN-13: 978-1449392925

Tutorial de python 3:
<https://docs.python.org/3/tutorial/>
<http://docs.python.org.ar/tutorial/3/index.html>

Title: Introducción a la programación con Python 3
By: Andrés Marzal Varó, Isabel Gracia Luengo, Pedro García Sevilla
Editor: Universitat Jaume I, 2014
ISBN: 978-84-697-1178-1
<http://repositori.uji.es/xmlui/bitstream/handle/10234/102653/s93.pdf>

Esta es la Guía Docente abreviada de la asignatura. Tienes también publicada en la Web la información más detallada de la asignatura en la Guía Docente Completa.