

MÁSTER EN ECONOMÍA, FINANZAS Y COMPUTACIÓN • 2017-2018

1. DATOS BÁSICOS DE LA ASIGNATURA

| | | | | | |
|-------------------------|-------------|-------------------------|----------|-----------------|-----|
| ASIGNATURA ESPAÑOL) | | PROGRAMACIÓN I | | | |
| SUBJECT | | COMPUTER PROGRAMMING I | | | |
| CÓDIGO | 1150201 | AÑO DE PLAN DE ESTUDIOS | | 2015 | |
| TIPO | OBLIGATORIO | X | OPTATIVO | | |
| MÓDULO | | PROGRAMACIÓN | | | |
| SEMESTRE | | 1º | | | |
| Créditos totales (ECTS) | 3,0 | TEORÍA (80%) | 18 | PRÁCTICAS (20%) | 4,5 |

HORARIO DE CLASES

| GRUPO | FECHA | Lunes | Martes | Miércoles | Jueves | Viernes |
|----------|------------|-------|---------------|-----------|---------------|---------|
| SESIÓN 1 | 14/11/2017 | | 16:30 a 20:30 | | | |
| SESIÓN 2 | 16/11/2017 | | | | 16:30 a 20:30 | |
| SESIÓN 3 | 28/11/2017 | | 16:30 a 20:30 | | | |
| SESIÓN 4 | 5/12/2017 | | 16:30 a 20:30 | | | |
| SESIÓN 5 | 12/12/2017 | | 16:30 a 20:30 | | | |
| SESIÓN 6 | 19/12/2017 | | 16:30 a 20:30 | | | |

2. DOCENTES

RESPONSABLE DE LA ASIGNATURA

| | | | | | |
|----------------------|---|-----------|---|----------|-----------------|
| NOMBRE | Iñaki Fernández de Viana y González (COORDINADOR) | | | | |
| UNIVERSIDAD | Huelva | | | | |
| DEPARTAMENTO | Tecnologías de la Información | | | | |
| ÁREA DE CONOCIMIENTO | Lenguajes y Sistemas Informáticos | | | | |
| Nº DESPACHO | 110 | UBICACIÓN | Edificio Torreumbría, Campus Universitario de la Rábida | | |
| CORREO ELECTRÓNICO | i.fviana@dti.uhu.es | | | TELÉFONO | +34.959.217.378 |
| URL WEB | www.uhu.es/i.fviana | | CAMPUS VIRTUAL | Moodle | |

HORARIO DE TUTORÍAS (*)

La celebración de las tutorías se comunicará a los alumnos a través de la plataforma virtual de la asignatura.

| | | | | | |
|----------------------|-------------------------------|-----------|---|----------|-----------------|
| NOMBRE | Gonzalo A. Aranda Corral | | | | |
| DEPARTAMENTO | Tecnologías de la Información | | | | |
| ÁREA DE CONOCIMIENTO | Inteligencia Artificial | | | | |
| Nº DESPACHO | 7 | UBICACIÓN | Edificio Torreumbría, Campus Universitario de la Rábida | | |
| CORREO ELECTRÓNICO | gonzalo.aranda@dti.uhu.es | | | TELÉFONO | +34-959.217.663 |
| URL WEB | www.uhu.es/gonzalo.aranda | | CAMPUS VIRTUAL | Moodle | |

HORARIO DE TUTORÍAS (*)

La celebración de las tutorías se comunicará a los alumnos a través de la plataforma virtual de la asignatura.

3. DESCRIPTOR

| | |
|---------|---|
| ESPAÑOL | <p>Los puntos básicos a tratar en la asignatura de Programación I se centran, inicialmente, en introducir los lenguajes de programación. Así, estudiaremos aspectos como el concepto e historia de los lenguajes de programación, la necesidad de aprender a programar, los paradigmas de programación y la diferencia entre compiladores e intérpretes. A continuación, introduciremos el concepto de programación orientada a objetos haciendo hincapié en los conceptos de objeto, clase, método y atributo. Seguiremos hablando de algoritmos y tipos de datos donde estudiaremos los tipos de algoritmos, tipos de datos, operadores, expresiones estructuras de control y tipos básicos de datos estructurados. Por último, hablaremos de distintos tipos de técnicas de diseño de programas, en concreto nos centraremos en el diseño descendente y modular.</p> |
|---------|---|

| | |
|----------------|---|
| ENGLISH | The course Programming 1 initially focuses on introducing programming languages. Thus, we will study issues such as the concept and history of programming languages, the need to learn programming, programming paradigms and the difference between compilers and interpreters. Then we introduce the concept of object-oriented emphasis on the concepts of object, class, method and attribute. We will continue talking about algorithms and data types: types of data and algorithms, operators, expressions, control structures and types of structured data. Finally, we will discuss about different types of program design techniques, in particular we will focus on the downstream and modular design. |
|----------------|---|

4. SITUACIÓN

PRERREQUISITOS

Ninguno.

CONTEXTO DENTRO DE LA TITULACIÓN

La asignatura de Programación I es la primera asignatura en la que el alumno conoce los lenguajes de programación de alto nivel para poder codificar programas. La asignatura tiene carácter obligatorio, impartándose en el segundo semestre del Máster.

RECOMENDACIONES

El alumno debe estudiar la asignatura consultando la bibliografía sugerida por los profesores y asistir con regularidad a las tutorías que estos ofertan. Debido al marcado carácter práctico de la asignatura, se recomienda que el alumnado realice la mayor parte de los ejercicios propuestos de las relaciones de problemas.

5. COMPETENCIAS

COMPETENCIAS

- CG1, CG10, CT1, CT2, CT3, CE5, CE8 Además el alumno debe adquirir: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería
- Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
- 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
- Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

6. OBJETIVOS DE LA ASIGNATURA

- Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas.
- Apreciar el papel central que representa la abstracción en la tarea de programar
- Desarrollar en el alumno la capacidad de resolución de problemas mediante las técnicas de diseño de algoritmos y aplicarlo a la codificación de programas
- Conocer y utilizar adecuadamente estructuras de datos básicas, algoritmos y esquemas de uso general
- Aprender un lenguaje de programación de alto nivel estructurado, general y extendido que use el paradigma de la programación orientado a objetos.
- Proporcionar los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación.

7. ACTIVIDADES FORMATIVAS, METODOLOGÍA DE ENSEÑANZA-APRENDIZAJE Y SU RELACIÓN CON COMPETENCIAS QUE DEBE ADQUIRIR EL ESTUDIANTE

| Metodología | Actividad | Descripción | Horas |
|---------------------------|---|---|-------|
| Clases presenciales | Clases teóricas | Clases teóricas. Fundamentos y planteamientos teóricos | 25,5 |
| | Clases prácticas | Problemas y casos prácticos: planteamiento y resolución de problemas concretos relacionados con la materia | |
| | | Laboratorio: desarrollo de habilidades de tecnología de la información y manejo de software específico de cada materia/asignatura | |
| Evaluación | Realización de exámenes parciales y finales, escritos u orales | | |
| Trabajo autónomo tutelado | Preparación de trabajos teórico-práctico individuales (prácticas planteadas y trabajo de curso) | Documentación Lectura Empleo de tecnologías de la información Resolución de problemas y preparación de casos | 46,5 |
| | Estudio y preparación de contenidos | Contenidos teóricos (conceptos y fundamentos) y prácticos de la materia | |
| | Lectura/s complementaria/s | Uso de manuales, monografías y artículos (científicos, didácticos y divulgativos) | |
| Tutorías | Individuales | Presenciales o no (Campus Virtual; correo electrónico). | 3 |

Todo el material oportuno para el seguimiento de las clases teóricas y prácticas estará disponible en la plataforma de teleformación Moodle (<http://moodle.uhu.es/>) Para las clases teóricas, los recursos que se utilizarán son la pizarra (tradicional y en su versión electrónica), las proyecciones de presentaciones con la ayuda del ordenador y material suplementario suministrado por el profesorado (fotocopias, archivos electrónicos, etc.). En las clases prácticas se aplicarán los contenidos abordados en las clases teóricas, se hará hincapié en los mecanismos de resolución, sus limitaciones y ventajas, así como un análisis crítico de los resultados alcanzados. Estas clases prácticas serán interactivas y la participación del/la alumno/a será tomada en cuenta a la hora de valorar su adaptación al grado de aprendizaje.

8. BLOQUES TEMÁTICOS Y CONTENIDOS

BREVE DESCRIPCIÓN DE LOS CONTENIDOS

Módulo I: INTRODUCCIÓN

- Lenguajes de programación
- Introducción a la programación
- Historia de los lenguajes de programación

Módulo II: Estructuras de datos y control de flujo

- Concepto de variable y tipo de dato
- Principales estructural de control de flujo
- Estructuras de datos: Listas, registros, tuplas, diccionarios, etc.

Módulo III: Programación orientada a objetos y gestión de entrada/salida

- Concepto de clase, objeto, método y atributo
- Lectura y escritura en fichero.
- Gestión de errores.

9. BIBLIOGRAFÍA

Los temas que se detallan en el programa utilizan cómo bibliografía las referencias que se indican a continuación. No obstante, a lo largo del curso se podrá incluir material adicional de lectura en el Aula Virtual para algunos temas, referencias que servirán como complemento y ampliación de lo explicado en clase y que serán también objeto de evaluación.

BIBLIOGRAFÍA GENERAL

- SHEPPARD, K.. Introduction to Python for Econometrics, Statistics and Data Analysis. University of Oxford. 2014.
- PILGRIM, M. Dive Into Python. Springer.2011.
- MARZAL, A y GRACIA I. Introducción a la programación con Python. Universidad Jaume I. 2010.

10. SISTEMA DE EVALUACIÓN

| Técnica empleada | Descripción | Criterio | Valor sobre el total de la nota |
|------------------|---|--|---------------------------------|
| | Prueba individual teórico-práctica, en la que se evaluará la adquisición conocimientos teórico-prácticos y metodológicos y diferentes técnicas de evaluación continua | <ul style="list-style-type: none"> •Grado de capacidad de resolución de problemas y aplicación de los contenidos teóricos a la práctica •Grado de conocimiento, comprensión e información •Ausencia de errores •Utilización adecuada de los conceptos •Capacidad de interrelacionar teorías, modelos, conceptos | 40% |
| | Prueba individual teórico-práctica al final de cada tema, en la que se evaluará la adquisición conocimientos teórico-prácticos y metodológicos y diferentes técnicas de evaluación continua | <ul style="list-style-type: none"> •Grado de capacidad de resolución de problemas y aplicación de los contenidos teóricos a la práctica •Grado de conocimiento, comprensión e información •Ausencia de errores •Utilización adecuada de los conceptos •Capacidad de interrelacionar teorías, modelos, conceptos | 30% |
| | Entrega de prácticas | <ul style="list-style-type: none"> •Grado de capacidad de resolución de problemas y aplicación de los contenidos teóricos a la práctica •Grado de conocimiento, comprensión e información •Ausencia de errores •Utilización adecuada de los conceptos •Capacidad de interrelacionar teorías, modelos, conceptos | 30% |

MEDIDAS PREVISTAS PARA RESPONDER A NECESIDADES EDUCATIVAS ESPECIALES

Se adoptarán, las medidas adecuadas a cada caso para que aquellos alumnos que presenten necesidades especiales puedan adquirir los conocimientos y capacidades necesarias para la superación de la materia.

D) ORGANIZACIÓN DEL TRABAJO DURANTE EL CURSO**11. NÚMERO DE HORAS DE TRABAJO DEL ESTUDIANTE**

Nº de Horas: 75 (3 créditos ECTS)

- Presencialidad total: 32 horas (Clases+Tutorías+Evaluación)
 - Clases teórico/prácticas: 24 horas -
 - Tutorías Especializadas (presenciales o virtuales):
Individuales: como medida indicativa el alumno debería invertir al menos diez horas en asistir a las tutorías del profesorado en el horario indicado o mediante consultas a través del correo electrónico: 5 horas
- Realización de Exámenes: 3 horas
- Otro Trabajo Personal Autónomo: 43 horas

E) TEMARIO DESARROLLADO**Tema 1. Introducción**

- 1.1. ¿Qué es la programación?
- 1.2. Historia de los lenguajes de programación
- 1.3. Paradigmas de programación
- 1.4. Compiladores e intérpretes
- 1.5. Entornos de desarrollo
- 1.6. Primer programa

Tema 2. Variables y tipos de datos

- 3.1 Variables
- 3.2 Tipos de datos
- 3.3 Operadores

Tema 3. Estructuras de control

- 3.1 Sentencias condicionales
- 3.2 Sentencias iterativas
- 3.3 Captura y tratamiento de excepciones
- 3.4 Funciones y módulos

Tema 4. Estructuras de datos

- 3.1 Listas
- 3.2 Diccionarios
- 3.3 Tuplas
- 3.5 Registros

Tema 5. Programación Orientada a Objetos

- 6.1 Definición de clases
- 6.2 Atributos y métodos
- 6.3 Encapsulación
- 6.4 Herencia

Tema 6. Gestión de entrada/salida

- 7.1 Ficheros
- 7.2 Directorios
- 7.3 Gestión de errores de Entrada/salida

F) MECANISMOS DE CONTROL Y SEGUIMIENTO DE LA ASIGNATURA

Cada vez que finalice cada tema del programa se realizarán pruebas diversas para establecer el nivel de captación de competencias y contenidos de la asignatura.