



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

GUÍA DOCENTE DE PROGRAMACIÓN I

La presente guía docente corresponde a la asignatura Programación I, aprobada para el curso lectivo 2017-2018 en Junta de Centro y publicada en su versión definitiva en la página web de la Escuela Politécnica Superior. La guía docente de Programación 1 aprobada y publicada antes del periodo de matrícula tiene el carácter de contrato con el estudiante.



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

ASIGNATURA

PROGRAMACIÓN I

1.1. Código

17817 del Grado en Ingeniería Informática

1.2. Materia

Informática

1.3. Tipo

Formación básica

1.4. Nivel

Grado

1.5. Curso

1º

1.6. Semestre

1º

1.7. Número de créditos

6 créditos ECTS

1.8. Requisitos previos

Esta asignatura se imparte en el primer semestre del primer curso, por lo que no se establece ningún requisito previo de programación.

Programación I forma parte de la Materia 1 del módulo de Programación y Estructuras de Datos del plan de estudios. Esta Materia está desglosada en tres asignaturas semestrales que se complementan entre sí: Programación I, Programación II y Proyecto de Programación (las dos últimas se imparten en el segundo semestre del primer curso). También se complementa con la asignatura Seminario-taller de



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

Software, del módulo Seminarios-taller de Informática, que se imparte simultáneamente a Programación I en el primer semestre. Por tanto, es imprescindible el buen aprovechamiento en cada una de estas asignaturas para superar con éxito todas ellas.

Se recomienda para garantizar la asimilación de los contenidos y la adquisición de habilidades la lectura crítica de los textos de la bibliografía, el uso del material electrónico de esta asignatura disponible en la plataforma Moodle (<http://moodle.uam.es>) y la búsqueda activa de material complementario en la red.

Es recomendable disponer de un cierto dominio del inglés escrito que permita al estudiante leer la bibliografía de consulta. Asimismo, es muy importante la disposición del estudiante para la realización de los ejercicios que cada día le propondrán los profesores. El estudiante deberá cotejar a diario su solución con la propuesta por el profesor con el fin de ir mejorando paulatinamente su estilo de programación.

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Esta asignatura dispone de dos métodos de evaluación: continua y no continua.

Es muy recomendable seguir el método de evaluación continua. Los estudiantes que se presenten a la prueba final de la convocatoria ordinaria serán evaluados por el método de evaluación no continua.

Los detalles de los métodos de evaluación se encuentran en la sección 4 de esta guía.

La asistencia a clase de teoría y prácticas es muy recomendable.

1.10. Datos del equipo docente

Nota: se debe añadir @uam.es a todas las direcciones de correo electrónico.

Coordinador de teoría:

Dr. Germán Montoro Manrique (Coordinador)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-410 Edificio B - 4ª Planta
Teléfono: +34 91 497 2210
Correo electrónico: german.montoro
Página web: <http://www.ii.uam.es/~montoro>

Horario de atención al alumnado: Petición de cita previa por correo electrónico.



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

1.11. Objetivos del curso

Programación 1 es una asignatura de introducción a la programación. Se pretende que el alumno adquiera unos conocimientos básicos sobre las técnicas de programación y la metodología del diseño de software aplicables a los lenguajes de alto nivel tradicionales. En particular, el alumno aprenderá a programar en el lenguaje C.

La **competencia** que se pretende adquirir con esta asignatura es:

B4: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Al final de cada unidad el estudiante deberá ser capaz de:

OBJETIVOS ESPECÍFICOS POR TEMA	
TEMA 1.- Introducción	
1.1.	Resolver problemas genéricos de programación
1.2.	Editar, compilar y enlazar un programa sencillo en un entorno de programación.
1.3.	Escribir información en la pantalla del ordenador.
1.4.	Leer información del teclado y guardarla en una variable.
1.5.	Depurar un programa con bucles y funciones.
TEMA 2.- Tipos de datos y operadores básicos	
2.1.	Declarar variables de tipos int, float y char.
2.2.	Trabajar con caracteres a través de su código ASCII.
2.3.	Utilizar los operadores de asignación, aditivos y multiplicativos.
2.4.	Utilizar paréntesis en las operaciones aritméticas.
TEMA 3.- Tablas, cadenas y estructuras	
3.1.	Distinguir una macro de una variable.
3.2.	Declarar tablas de tipos básicos, cadenas y estructuras.
3.3.	Acceder a los elementos de una tabla unidimensional y bidimensional.
3.4.	Acceder a los miembros de una variable estructurada.
3.5.	Acceder a los miembros de tablas de estructuras.
3.6.	Acceder a los miembros de una variable de un tipo estructurado anidado.
TEMA 4.- Instrucciones de control	
4.1.	Escribir y evaluar operaciones lógicas sencillas.
4.2.	Distinguir la instrucción if-else de la instrucción switch.
4.3.	Utilizar enumeraciones combinadas con instrucciones switch.
4.4.	Transformar un bucle for en un bucle while equivalente.
4.5.	Utilizar bucles for para leer y escribir los elementos de una tabla.
4.6.	Utilizar bucles anidados para navegar por tablas bidimensionales.
TEMA 5.- Funciones y punteros	
5.1.	Definir y llamar funciones con argumentos.



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

5.2.	Declarar punteros.
5.3.	Asignar la dirección de una variable a un puntero.
5.4.	Acceder a la variable apuntada desde un puntero.
5.5.	Distinguir un argumento pasado por valor de uno pasado por referencia.
5.6.	Recorrer los caracteres de una cadena mediante un puntero.
5.7.	Reservar memoria dinámica para tablas unidimensionales.
5.8.	Utilizar cadenas de caracteres como argumentos y las funciones de string.h.
5.9.	Acceder a los campos de una variable estructurada desde su dirección.
5.10.	Pasar estructuras por valor y por referencia.
5.11.	Pasar tablas y cadenas como argumentos.
TEMA 6.- Archivos de texto	
6.1.	Leer un archivo de texto carácter a carácter y palabra a palabra.
6.2.	Leer un archivo de texto con datos numéricos.
6.3.	Escribir un archivo de texto carácter a carácter.
TEMA 7.- Estructura de un programa	
7.1.	Crear archivos de cabecera.
7.2.	Repartir un programa entre varios archivos.
7.3.	Diseñar un programa de forma descendente.

1.12. Contenidos del programa

Programa Sintético

TEMA 1. Introducción
TEMA 2. Tipos de datos y operadores básicos
TEMA 3. Tablas, cadenas y estructuras
TEMA 4. Instrucciones de control
TEMA 5. Funciones y punteros
TEMA 6. Archivos de texto
TEMA 7. Estructura de un programa

Programa Detallado

1. **Introducción**
 1. Edición, compilación y enlazado de un programa
 2. Escritura de información
 3. Lectura de información

2. **Tipos de datos y operadores básicos**
 1. Tipos atómicos
 2. El código ASCII
 3. Operadores básicos

3. **Tablas, cadenas y estructuras**
 1. Macros



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

2. Introducción a las tablas
 3. Introducción a las cadenas
 4. Introducción a las estructuras
 5. Estructuras anidadas
 6. Tablas de estructuras
- 4. Instrucciones de control**
1. Operadores relacionales y de igualdad
 2. La instrucción if-else
 3. La instrucción switch y las enumeraciones
 4. El bucle while
 5. El bucle do-while
 6. El bucle for
 7. Bucles anidados
- 5. Funciones y punteros**
1. Funciones sin argumentos
 2. Funciones con argumentos
 3. Alcance de las variables
 4. Punteros
 5. Paso de argumentos
 6. Punteros y tablas
 7. Paso de tablas
 8. Paso de cadenas
 9. Paso de estructuras
 10. Reserva dinámica de memoria
- 6. Archivos de texto**
1. Lectura de archivos de texto
 2. Escritura de archivos de texto
- 7. Estructura de un programa**
1. Archivos de cabecera
 2. Proyectos con más de un archivo
 3. Makefile
 4. Diseño descendente

1.13. Referencias de consulta

Bibliografía:

A continuación se listan algunos libros de texto ordenados por su nivel de dificultad. Conviene que el estudiante consulte estas obras en la biblioteca antes de decidirse a comprar alguna de ellas.

Nivel básico:



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

1. [C Programming: A modern approach](#) (second edition). K. N. King. W. W. Norton and Company, 2008. Excelente manual de C, bien estructurado, actualizado con el estándar C99 y accesible no solo para principiantes en C, sino incluso para programadores noveles. Se trata de uno de los textos de C de referencia en muchas universidades norteamericanas.
2. [C. Primer Plus](#) (sixth edition). Stephen Prata. Sams Publishing, 2013. Este es un libro que explica con mucha claridad los cambios experimentados por el lenguaje de un estándar a otro, incluidos C99 y C11. Cada capítulo contiene cuestiones de repaso y ejercicios de programación.
3. [Head first C](#). David Griffiths. O'Reilly, 2013. Como otros libros de la colección *Head First*, este texto de C utiliza los últimos resultados sobre procesos de aprendizaje para facilitar el aprendizaje del lenguaje de programación C. Su lectura resulta muy entretenida y seguro que sorprende al estudiante.
4. [A book on C. Programming in C](#) (fourth edition). Kelley, Al and Pohl, Ira. Addison-Wesley, 2004. Este texto utiliza el método de la disección, ideado por el autor en 1984, que no es otra cosa que una descripción detallada de cada programa instrucción a instrucción. La última edición contiene material para facilitar el tránsito del estudiante a C++ y Java.
5. [Introducción a la computación](#) (edición 12ª). Brookshear, J. Glenn. Pearson Addison-Wesley, 2013. Este libro es una introducción general y muy accesible a la informática. En el capítulo primero se estudia la representación de la información como cadenas de bits.

Nivel medio:

6. [Programación en C/C++](#) (tercera edición). Alejandro Sierra, Manuel Alfonseca. Anaya Multimedia, 2014. Este texto es una guía rápida que describe conjuntamente las características de los lenguajes C y C++, destacando las diferencias de enfoque entre ambos lenguajes. En la última edición se han incorporado las principales novedades de los estándares C99, C11 y C++11. Puede resultar muy útil para saltar a C++ una vez que se disponga de conocimientos básicos sobre C.
7. [Introductory C: pointers, functions and files](#). Richard Petersen. Academic Press, 1997. Este libro pone especial énfasis en todo aquello que tiene que ver con punteros. Está especialmente indicado para aquellos programadores que temen a los punteros. Cada capítulo termina con un interesante resumen.
8. [La práctica de la programación](#). B.W. Kernighan. Pearson Educación, 2000. Se trata de un texto compacto que responde de forma práctica a cuestiones relacionadas con el estilo, diseño, depuración, pruebas, eficiencia y mantenimiento del software. Los ejemplos están escritos en los lenguajes C y C++.

Nivel avanzado:

9. [The C programming language](#). Brian W. Kernighan, Dennis M. Ritchie. Prentice Hall, 1988. Es sin duda el libro de programación en C más famoso de la historia y está escrito por los diseñadores del lenguaje. Su influencia ha sido tal que hoy en día el primer ejemplo de casi cualquier lenguaje es un ejemplo ideado por los autores: el programa hola mundo. Otra característica que han incorporado muchos textos posteriores es su capítulo 0 que contiene una



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

rápida introducción al lenguaje en menos de 30 páginas. Es una referencia indispensable para cualquier programador que se recomienda adquirir en su idioma original.

10. [Ingeniería de software clásica y orientada a objetos](#) (sexta Edición). Stephen Schach. McGraw-Hill. Este libro que ya es un clásico presenta una introducción amigable a la ingeniería del software tanto clásica como orientada a objetos.

Material electrónico de trabajo: los documentos electrónicos de trabajo (resumen de comandos, recomendaciones sobre la elaboración de documentación y pseudocódigo, recomendaciones sobre legibilidad en el código, tabla ASCII, ejercicios del curso y ejemplos de exámenes, etc.) se publican en la sección de Programación 1 de la plataforma Moodle (<https://moodle.uam.es>)

2. Métodos docentes

Todas las clases tendrán un fuerte contenido práctico y en su mayoría se celebrarán en el laboratorio. Lo que designamos como clases de teoría en el apartado 3 de esta guía corresponden a las 10 horas de explicaciones repartidas a lo largo del curso y que consistirán en breves introducciones a algunos temas como, por ejemplo, el entorno de programación al principio del curso.

Las clases comenzarán con una breve exposición práctica por parte del profesor, las más de las veces mediante el cañón de proyección. Cada sesión introducirá un nuevo ingrediente del lenguaje. Seguidamente los estudiantes dedicarán un periodo de tiempo a la visualización de un vídeo explicativo correspondiente a la sesión y, finalmente, realizarán los ejercicios de programación propuestos por el profesor. La clase será interrumpida por el profesor para poner en común comentarios relevantes sobre la ejecución de los ejercicios.

En las sesiones de laboratorio, los estudiantes tendrán que completar una prueba de evaluación sobre el trabajo realizado hasta la semana anterior. La calificación continua de la asignatura corresponderá a la media ponderada de estas pruebas. Los estudiantes que acudan a la prueba final serán evaluados exclusivamente por el resultado de la prueba final.

3. Tiempo de trabajo del estudiante

		Nº de horas	Porcentaje
Presencial	Clases teórico-prácticas	70 h (47%)	82 h (55%)
	Tutorías	6 h (4%)	
	Realización de pruebas finales	6 h (4%)	
No presencial	Estudio semanal	36 h (24%)	68 h (45%)
	Preparación de pruebas finales	32 h (21%)	
Carga total de horas de trabajo: 25 horas x 6 ECTS		150 h	



Asignatura: Programación I
Código: 17817
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación básica
Nº de créditos: 6

4. Métodos de evaluación y porcentaje en la calificación final

Existen dos métodos de evaluación en esta asignatura: continua y no continua.

El estudiante que se presente a la prueba final será evaluado por el método de evaluación no continua.

Es imprescindible que el estudiante lea atentamente la normativa de evaluación de la EPS y de la UAM puesto que se aplicarán con rigor, concretamente a las copias.

Método de evaluación continua

- Los estudiantes serán calificados mediante 5 pruebas prácticas (P_1, \dots, P_5) y 2 pruebas teóricas (T_1 y T_2).
- Las pruebas son individuales.
- Cada prueba se calificará entre 0 y 10.
- El estudiante que no se presente a una prueba será calificado con un 0 en la prueba correspondiente.
- La calificación será igual a la media ponderada según la siguiente fórmula:

$$C = (P_1 + P_2 + P_3 + T_1 + 2 \cdot P_4 + 3 \cdot T_2 + 3 \cdot P_5) / 12$$

Si el estudiante se presenta a menos de 3 pruebas recibirá la calificación "No evaluado". En caso contrario recibirá una calificación numérica correspondiente a la media de las pruebas.

Método de evaluación no continua.

- La calificación final de la asignatura será la correspondiente a una prueba global.

Convocatoria extraordinaria.

- La calificación de la convocatoria extraordinaria será la correspondiente a una prueba global.

5. Cronograma

Semana	Contenido	Horas presenciales	Horas no presenciales
1	- Presentación de la asignatura apoyada por la guía docente. Descripción de la plataforma Moodle. TEMA 1. Introducción	3 - Editar, compilar, enlazar y ejecutar	3 - Leer guía docente. - Darse de alta en Moodle



Asignatura: Programación I
 Código: 17817
 Centro: Escuela Politécnica Superior
 Titulación: Grado en Ingeniería Informática
 Nivel: Grado
 Tipo: Formación básica
 Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
	- 1.1 Edición, compilación y enlazado de un programa - 1.2 Escritura de información - 1.3 Lectura de información	un programa de saludo. - Ejercicios.	y UAMx e inscribirse en la asignatura. - Familiarizarse con el entorno de programación. - Ver vídeos del TEMA 1
2	TEMA 2. Tipos de datos y operadores básicos - 2.1 Tipos atómicos - 2.2 El código ASCII - 2.3 Operadores básicos	5 - Ejercicios - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 2. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
3	TEMA 3. Tablas, cadenas y estructuras - 3.1 Macros - 3.2 Introducción a las tablas - 3.3 Introducción a las cadenas - 3.4 Introducción a las estructuras - 3.5 Estructuras anidadas - 3.6 Tablas de estructuras	5 - Ejercicios - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 3. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
4	TEMA 4. Instrucciones de control - 4.1 Operadores relacionales y de igualdad - 4.2 La instrucción if-else - 4.3 La instrucción switch y las enumeraciones	5 - Ejercicios - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 4. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
5	TEMA 4. Instrucciones de control - 4.4 El bucle while - 4.5 El bucle do-while Festivo: jueves, 12 de octubre	3 - Ejercicios. - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 4. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
6	TEMA 4. Instrucciones de control - 4.6 El bucle for - 4.7 Bucles anidados PRUEBA NO EVALUABLE.	5 - Ejercicios. - Trabajar en laboratorio. - Prueba no evaluable para familiarizarse con el entorno.	3 - Ver vídeos del TEMA 4. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
7	EJERCICIOS AVANZADOS - Ejercicios avanzados de repaso del temario visto hasta el momento PRUEBA PRÁCTICA 1.	5 - Ejercicios. - Trabajar en laboratorio. - Prueba práctica de evaluación 1	3 - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
8	TEMA 5. Funciones y punteros - 5.1 Funciones sin argumentos - 5.2 Funciones con argumentos - 5.3 Alcance de las variables	3	3 - Ver vídeos del TEMA 5. - Hacer ejercicios propuestos en clase y/o



Asignatura: Programación I
 Código: 17817
 Centro: Escuela Politécnica Superior
 Titulación: Grado en Ingeniería Informática
 Nivel: Grado
 Tipo: Formación básica
 Nº de créditos: 6

Semana	Contenido	Horas presenciales	Horas no presenciales
	Festivo: miércoles, 1 de noviembre		comparar la solución de los ejercicios con la propuesta por el profesor.
9	TEMA 5. Funciones y punteros - 5.4 Punteros - 5.5 Paso de argumentos por referencia - 5.6 Punteros y tablas Festivo: jueves, 9 de noviembre	3 - Ejercicios. - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 5. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
10	TEMA 5. Funciones y punteros - 5.7 Paso de tablas - 5.8 Paso de cadenas - 5.9 Paso de estructuras PRUEBA PRÁCTICA 2.	5 - Ejercicios. - Trabajar en laboratorio. - Prueba práctica de evaluación 2.	3 - Ver vídeos del TEMA 5. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
11	TEMA 5. Funciones y punteros - 5.10 Reserva dinámica de memoria PRUEBA TEÓRICA 1.	5 - Ejercicios. - Trabajar en laboratorio. - Prueba teórica de evaluación 1.	3 - Ver vídeos del TEMA 5. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
12	EJERCICIOS AVANZADOS - Ejercicios avanzados de repaso del temario visto hasta el momento PRUEBA PRÁCTICA 3.	5 - Ejercicios. Trabajar en laboratorio. - Prueba práctica de evaluación 3.	3 - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
13	TEMA 6. Archivos de texto - 6.1 Lectura de archivos de texto - 6.2 Escritura de archivos de texto Festivo: miércoles, 6 de diciembre	3 - Ejercicios. - Trabajar en laboratorio.	3 - Ver vídeos del TEMA 6. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
14	TEMA 7. Estructura de un programa - 7.1 Archivos de cabecera - 7.2 Proyectos con más de un archivo - 7.3 Makefile - 7.4 Diseño descendente PRUEBA PRÁCTICA 4.	5 - Ejercicios. - Trabajar en laboratorio. - Prueba práctica de evaluación 4.	3 - Ver vídeos del TEMA 7. - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.
15	PRUEBA TEÓRICA 2 PRUEBA PRÁCTICA 5	5 - Prueba teórica de evaluación 2. - Prueba práctica de evaluación 5.	3 - Hacer ejercicios propuestos en clase y/o comparar la solución de los ejercicios con la propuesta por el profesor.