



PROGRAMA DE LA ASIGNATURA

Curso académico 2008-2009

Identificación y características de la asignatura				
Denominación	PROGRAMACION II		Código	104609
Créditos (T+P)	3 de teoría + 4,5 de práctica			
Titulación	INGENIERÍA TÉCNICA EN TELECOMUNICACIÓN - ESPECIALIDAD EN TELEMÁTICA			
Centro	Centro Universitario de Mérida			
Curso	2	Temporalidad	1ºC	
Carácter	Troncal			
Descriptor(es) (BOE)	Práctica de desarrollo de programas. Pruebas funcionales. Otros tipos de lenguajes			
Profesor/es	Nombre	Despacho	Correo-e	Página web
	Luis J. Arévalo Rosado	9	ljarevalo@unex.es	CampusVirtual
Área de conocimiento	LENGUAJES Y SISTEMAS INFORMÁTICOS			
Departamento	INGENIERÍA DE SISTEMAS INFORMÁTICOS Y TELEMÁTICOS			
Profesor coordinador (si hay más de uno)				

UNIVERSIDAD DE EXTREMADURA Centro Universitario Mérida
ENTRADA: 032372
09/07/2008 11:53:19 (8433170)



Objetivos y/o competencias

Objetivos:

Descripción	Vinculación (CET)
1. Afianzar al alumno en el desarrollo de programas informáticos basados en la metodología de programación estructurada.	13,14
2. Iniciar al alumno a la programación orientación a objetos	13,14
3. Permitir a los alumnos resolver supuestos que supongan la implementación de programas de tamaño medio, con codificaciones correctas y eficientes, utilizando el paradigma de la programación estructurada junto con la orientada a objetos.	13,14
4. Preparar al alumno para la implementación de programas de gran tamaño y complejidad.	13,14
5. Presentar las estructuras de datos lineales básicas y orientar al alumno para que sea capaz de seleccionar la más adecuada en cada problema.	13,14
6. Presentar las estructuras de datos no lineales básicas y orientar al alumno para cuando debe usarse.	13,14
7. Capacitar al alumno para el tratamiento de información almacenada en dispositivos de almacenamiento secundario	13,14

Competencias

Descripción	Vinculación (CET)
8. Adquirir la capacidad de analizar, diseñar e implementar en un desarrollo software las necesidades para poder resolver un problema informático.	1,9
9. El alumno debe conocer los principales conceptos de un lenguaje de programación orientado a objetos.	1,9
10. Desarrollar la capacidad de dimensionar y relativizar los problemas, aplicando resultados menores en la consecución de proyectos más ambiciosos.	1,9
11. Trabajar en equipo	9,12

Temas y contenidos

(especificar prácticas, teoría y seminarios, en su caso)

**TEMA 1. Punteros y referencias**

- 1.Introducción
- 2.Apuntadores
- 3.Fundamentos
- 4.Operadores
 - 1.Operaciones Básicas
 - 2.Otras características
 - 3.Punteros y arrays
 - 4.Punteros y estructuras
- 5.Gestión dinámica de memoria
- 6.Referencias
- 7.Bibliografía

TEMA 2. Programación orientada a objetos

- 1.Introducción
- 2.Declaración de una clase
- 3.Encapsulación
- 4.Representación gráfica de una clase
- 5.Constructores y destructores
- 6.Creación y eliminación dinámica de objetos
- 7.Paso y retorno de objetos a funciones
- 8.Otras características
 - 1.Operador asignación
 - 2.Operador this
 - 3.Atributos estáticos
 - 4.Métodos constantes
- 9.Bibliografía
- 10.Ejemplos y ejercicios

TEMA 3. Sobrecarga de operadores

- 1.Introducción. Concepto de sobrecarga.
- 2.Sobrecarga de funciones.
- 3.Sobrecarga de métodos.
- 4.Sobrecarga de operadores.
 - 1 . Sobrecarga de operadores unarios.
 - 2 . Sobrecarga de operadores binarios.
 - 3 . Sobrecarga como métodos y como funciones amigas.
- 5.Sobrecarga del operador <<
- 6.Ejemplos.

TEMA 4. Herencia y polimorfismo

- 1.Introducción a la Herencia
 - 1.Tipos de herencia
 - 1.Herencia simple
 - 2.Herencia múltiple
 - 3.Herencia privada
 - 4.Herencia pública
 - 5.Herencia protegida
 - 2.Constructores en herencia



3. destructores en herencia
 4. Métodos: Sobrecarga vs Redefinición
 5. Asignación de instancias
 6. Ejercicios y problemas
2. Introducción al polimorfismo
 1. Punteros a clases derivadas
 2. Funciones virtuales
 3. Funciones virtuales puras
 4. Clases abstractas
 5. Vinculación dinámica
 6. Ejercicios y problemas

TEMA 5. Excepciones

1. Introducción
2. Estructura de excepciones
3. Excepciones definidas por el usuario
4. Clase "exception"
5. Ejercicios y problemas

TEMA 6. Estructura de datos lineales

1. Introducción
2. TAD Pila
 1. Descripción
 2. Especificación algebraica
 3. Especificación de la clase Pila
 4. Implementaciones de la clase Pila
3. TAD Cola
 1. Descripción
 2. Especificación algebraica
 3. Especificación de la clase Cola
 4. Implementaciones de la clase Cola
4. TAD Lista
 1. Descripción
 2. Especificación algebraica
 3. Especificación de la clase
 4. Implementaciones de la clase Lista
5. Ejercicios y problemas

TEMA 7. Estructuras de Datos Avanzadas: Árboles

1. Introducción
2. TAD Arbol
3. TAD Arbol Ordenado
4. Otros conceptos
5. TAD Árbol binario
6. Ejercicios y problemas

TEMA 8. Ficheros

1. Introducción
2. Operaciones básicas con ficheros



- 3.Apertura y cierre
- 4.Lectura y Escritura
- 5.Acceso aleatorio
- 6.Errores de entrada / salida
- 7.Ejercicios y problemas

TEMARIO DE LA PARTE PRÁCTICA

- Práctica 1: Repaso e introducción al lenguaje C++
 Práctica 2: Gestión de memoria dinámica. Punteros.
 Práctica 3: Clases I.
 Práctica 4: Clases II. Composición.
 Práctica 5: Herencia
 Práctica 6: Polimorfismo.
 Práctica 7: EDA Lineales I
 Práctica 8: EDA Lineales II
 Práctica 9: Árboles
 Práctica 10: Otras características de C++

Notas adicionales:

- Durante el curso se realizarán distintos crucigramas y cuestionarios sobre la materia con el objetivo de ayudar al alumnado en su estudio diario.

Actividades de Enseñanza-Aprendizaje

Descripción	Modal.	Tipo	Duración	Temas	Objetivos
1. Presentación de la asignatura	GG	C-E	0,5 h	1-7	
2. Encuesta de conocimientos previos	GG	C-E	0,5 h		
3. Práctica 0. Repaso de programación imperativa	S	P	3 h		1, 3
4. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 1	NP	T	1 h	1	1, 3
5. Exposición sobre conceptos de programación dinámica (Tema 1)	GG	T	3 h	1	1, 3
6. Estudio de los contenidos explicados en el tema 1	NP	T	2 h	1	1, 3
7. Práctica 1. Programación dinámica	S	P	3 h	1	1, 3
8. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 2	NP	T	2 h	2	2-4



9. Exposición sobre conceptos de programación orientada a objetos (Tema 2)	GG	T	3 h	2	2-4
10. Estudio de los contenidos explicados en el tema 2	NP	T	3 h	2	2-4
11. Resolución de problemas de pizarra sobre POO (Tema 2)	GG	T-P	2 h	2	2-4
12. Práctica 2. Programación orientada a objetos	S	P	3 h	2	2-4
13. Desarrollo y entrega de un caso práctico de POO	NP	P	3 h	2	2-4
14. Tutoría ECTS	Tut	T-P	1 h	1, 2	1-4
15. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 3.	NP	T	2 h	3	2-4
16. Exposición sobre conceptos de sobrecarga. Tema 3.	GG	T	2 h	3	2-4
17. Estudio de los contenidos explicados en el tema 3	NP	T	2 h	3	2-4
18. Práctica 3. Sobrecarga y Composición	S	P	3 h	2, 3	2-4
19. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 4.	NP	T	2 h	4	2-4
20. Exposición sobre conceptos de herencia y polimorfismo (Tema 4)	GG	T	2 h	4	2-4
21. Estudio de los contenidos explicados en el tema 4	NP	T	2 h	4	2-4
22. Resolución de problemas de pizarra sobre herencia y polimorfismo (tema 4)	GG	T-P	2 h	4	2-4
23. Práctica 4. Herencia y polimorfismo	S	P	5 h	4	2-4
24. Desarrollo y entrega de un caso práctico sobre herencia y polimorfismo	NP	P	4 h	4	2-4
25. Tutorías ECTS	Tut	T-P	1 h	1-5	1-4
26. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 5.	NP	T	1 h	5	2-4
27. Exposición sobre conceptos de excepciones	GG	T	2 h	5	2-4



28. Estudio de los contenidos explicados en el tema 5	NP	T	1 h	5	2-4
29. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 6.	NP	T	2 h	5	2, 4, 5
30. Exposición sobre conceptos de Estructuras de Datos Lineales	GG	T	4 h	6	2, 4, 5
31. Estudio de los contenidos explicados en el tema 6	NP	T	4 h	6	2, 4, 5
32. Resolución de problemas de pizarra sobre el Est. Datos Lineales	GG	T-P	3 h	6	2, 4, 5
33. Práctica 5. Est. Datos Lineales	S	P	7 h	6	2, 4, 5
34. Desarrollo y entrega de un caso práctico sobre Est. Datos Lineales	NP	P	6 h	6	2, 4, 5
35. Tutorías ECTS	Tut	T-P	1 h	6, 7	1-5
36. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 7.	NP	T	2 h	7	2, 4, 6
37. Exposición sobre conceptos de Estructuras de Datos no Lineales	GG	T	4 h	7	2, 4, 6
38. Estudio de los contenidos explicados en el tema 7	NP	T	5 h	7	2, 4, 6
39. Resolución de problemas de pizarra sobre el Est. Datos no Lineales	GG	T-P	3 h	7	2, 4, 6
40. Práctica 6. Est. Datos no Lineales	S	P	3 h	7	2, 4, 6
41. Desarrollo y entrega de un caso práctico sobre Est. Datos no Lineales	NP	P	6 h	7	2, 4, 6, 8
42. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 8.	NP	T	2 h	8	2, 4
43. Exposición sobre conceptos de ficheros	GG	T	2 h	8	2, 4
44. Estudio de los contenidos explicados en el tema 8	NP	T	2 h	8	2, 4
45. Práctica 7. Resolución de dudas	S	P	1 h	8	1-7
46. Desarrollo y entrega de un caso práctico final	NP	P	11 h	1-8	1-7



47. Elaboración de memorias de prácticas de laboratorio	NP	T-P	4 h		8
48. Resolución de dudas	GG	T	2 h	1-8	
49. Asistencia a tutorías complementarias para aclaración de dudas	NP	T-P	3 h	1-8	1-7
50. Preparación del examen final	NP	T	9 h	1-8	1-7
51. Examen Final	GG	C-E	3 h	1-8	1-7

Modalidad: GG = Grupo grande; S = Seminario - Laboratorio; Tut = Tutoría ECTS; NP = No presencial
 Tipo: C-E = Coordinación / evaluación; T = Teórica; P = Práctica; T-P = Teórica / práctica

Criterios de evaluación

- La materia se divide en dos partes, una parte teórica y otra práctica.

Teoría:

- Se realizará un examen en la convocatoria oportuna formado por dos apartados.
- El primer apartado constará de preguntas tipo test o preguntas cortas. El objetivo de este examen es que el alumno demuestre que ha adquirido los conocimientos teóricos de la asignatura. Las preguntas falladas restarán con respecto a la nota final. Las preguntas dejadas en blanco (no contestadas) no puntuarán ni positiva ni negativamente a la hora del computo final de la nota. Este examen tendrá una nota comprendida entre 10%-15% de la nota de teoría.
- El segundo apartado consistirá en la resolución de varios problemas de programación. Este examen tendrá un peso del 45% de la nota de final.
- El tiempo del examen será determinado por el profesor de la asignatura dependiendo del número de preguntas del examen. En su defecto y a petición de un número de alumnos superior al 50% del número de alumnos que se encuentran realizando el examen, el profesor asignado para la vigilancia del examen podrá tomar la decisión de aumentar la duración del examen en 15 minutos.
- La nota de teoría supone el 50% de la nota final de la asignatura.

Prácticas:

- Se realizarán varios ejercicios prácticos a lo largo del curso con el objetivo que el alumno adquiera suficiente soltura para implementar la práctica final de la asignatura.
- Esta práctica final deberá entregarse en la fecha indicada por el profesor.
- Se deberá realizar una defensa de la práctica para demostrar la autoría de la misma. Esta defensa será o bien una defensa oral o bien un examen práctico, en cuyo caso consistirá en una modificación de la



práctica que se deberá realizar correctamente para aprobar la parte práctica.

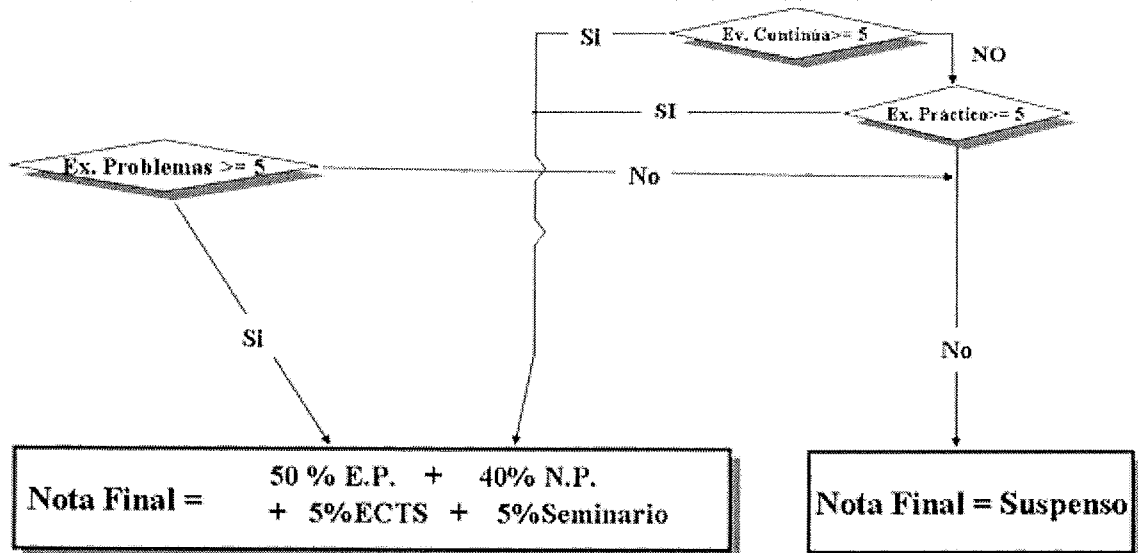
- Con el objetivo que el alumno vaya realizando la práctica paulatinamente, se solicitarán tres entregas de la misma.
- En la convocatoria de Junio y Septiembre la evaluación de las prácticas será mediante examen práctico, debido a que no se puede realizar un seguimiento de práctica.
- Cualquier sospecha de copia sobre una práctica entregada o parte de la misma, implicará inexorablemente suspender la práctica completa, tanto a la persona que la copió cómo a la persona que permitió copiarla.
- Tanto la nota del examen teórico como la nota de la parte práctica podrán ser guardadas hasta la convocatoria de septiembre de este curso, siempre que sea superior a 5.
- La nota de práctica supone el 40% de la nota final de la asignatura.

Nota

- Para aprobar el curso completo ha de obtenerse una nota mínima de 5 en el examen de teoría, así como tener aprobadas las prácticas.

TEORÍA

PRÁCTICA





Bibliografía

Bibliografía de apoyo seleccionada

1. Programación orientada a objetos. Roberto Rodríguez, Encarna Sosa y Alvaro Prieto.
2. Programación en C++. Luis Joyanes.
3. Como programar en C++ . H. M. Deitel.
4. Resolución de problemas con C++. Savitch.
5. El lenguaje de programación C++. Stroustrup, B.
6. Aprendiendo C++ para linux en 21 días. Jesse Liberty y David B. Horvath.
7. Thinking in C++. Bruce Eckel.
8. Estructura de Datos, Algoritmos y Programación Orientada a Objetos. Gregory L. Heileman. Ed. McGraw-Hill, 1998
9. Fundamentals of Data Structures in C++. E. Horowitz, S. Sahni, D. Mehta. Computer Science Press, 1997

*Bibliografía o documentación de lectura obligatoria**

10. Programación orientada a objetos. Roberto Rodríguez, Encarna Sosa y Alvaro Prieto.
11. Aprendiendo C++ para linux en 21 días. Jesse Liberty y David B. Horvath.
12. Problemas resueltos de programación en lenguaje C++. Ed. Thomson Paraninfo. García Sánchez, J.D y otros.

*Bibliografía o documentación de ampliación, sitios web...**

13. Thinking in C++. Bruce Eckel. <http://www.mindview.net/Books/TICPP/>
14. Zator: <http://www.zator.com/Cpp/>
15. Aprende C++ como si estuviera en primero. Manual en PDF.

Tutorías

Luis J. Arévalo Rosado

- Tutoría Presencial. Despacho nº 9
 - Miércoles de 16:00 a 18:00
 - Jueves de 9:00 a 11:00
 - Viernes de 10:00 a 12:00
- Tutoría no Presencial
 - MSN: prof_luis_are@hotmail.com