

科目名	ソフトウェア設計論Ⅱ Software Design and Development II			担当教員	金澤啓三		
学年	3年	学期	後期	履修条件	必修	単位数	2
分野	専門	授業形式	講義・演習	科目番号	15237013	単位区別	履修
学習目標	C, C++言語による各種抽象的なデータ構造を学習し、構造化プログラミング、オブジェクト指向プログラミングを習得することにより応用的なプログラミング能力を養成する。プログラミング演習はLinuxのパーソナルコンピュータ上で行い、課題プログラムの演習を通じて、設計からデバッグまで将来に渡って様々なシーンで活用できる実践的なプログラミング能力を身に付けることを目標とする。						
進め方	学習項目ごとに、学習内容の解説と関連するプログラムの例題、演習課題を講義する。その後、演習により課題のプログラムを作成し、レポートとして提出する。プログラミング言語にはC言語及びC++言語を用いる。後半にはオブジェクト指向プログラミングをグループ単位で行い、小規模なシステム開発をする。また、理解度確認のための小テストを不定期に実施する。						
学習内容	学習項目（時間数）			学習到達目標			
	1. ガイダンス (2)			構造体を理解し、提示されたプログラム課題に活用できる。 D2:2, E2:2, E3:2 オブジェクト指向の基本的な概念を理解する。 D2:1 クラスの基本的な概念、機能を理解し、クラスの定義ができる。 D2:2, E2:2, E3:2			
	2. 構造体とポインタ (10) (1) 構造体へのポインタ (2) 自己参照構造体 (3) 連結リスト						
	3. オブジェクト指向 (4) (1) オブジェクト指向論 (2) 開発環境						
4. クラス① (12) (1) カプセル化と情報隠蔽 (2) クラスとインスタンス (3) メンバとメソッド (4) コンストラクタとデストラクタ							
[後期中間試験] (2)							
5. 試験問題の解答 (2)			クラスの継承し、機能拡張を行う手法について理解する。 D2:2, E2:2, E3:2 例外処理を利用することができる。 D2:2, E2:2, E3:2 クラスを活用し、簡単なシステム設計や実装ができる。 D2:2, E2:1-3, E3:1-3 プログラム要素の再利用性を高める手法について理解する。 D2:1				
6. クラス② (6) (1) 継承とポリモーフィズム (2) 抽象クラス							
7. 例外処理 (4)							
8. クラスの設計演習 (12) (1) 要求分析 (2) システム分析 (3) 詳細設計 (4) テスト駆動開発							
9. 再利用技術 (4)							
後期末試験							
10. 試験問題の解答 (2)							
11. 相互評価 (2)							
評価方法	定期試験 70%, 演習レポート・提出物を 30%の比率で評価する。						
履修要件	特になし						
関連科目	情報処理 I (2年), 基礎工学実験・実習 (2年), ソフトウェア設計論 I (3年)						
教材	教科書: プリント配布						
備考	オフィスアワー: 毎金曜日放課後~17:00						