

授業科目名		コンピュータとソフトウェアの基礎			
単位数		2			
開設学期		1年生前期			
目的		コンピュータの構成と動作原理やシステムの基本的な概念、さらに、コンピュータ上で動作するソフトウェアおよびその開発技術の概念と意義について理解することを目的とする。			
概要		コンピュータの概念、基本構造、データのメモリ上での表現、アセンブリ言語の構成、メモリシステム、入出力インタフェースといった一連のコンピュータの構造とシステムの基本的な概念、ならびにソフトウェア開発の全体像を学ぶ。			
目標		コンピュータの構成と動作原理、システムの基本的な概念、ソフトウェアおよびその開発技術の基本的な概念を習得する			
先修科目					
関連科目					
授業方法					
評価方法・基準					
授業展開		学習成果	トピック	学習目標	発展学習目標
1	SEコースを概説し、工学的アプローチについて述べる。また本講義の概要と目的および進め方について説明する(※講義0.3h)。		コースの全体説明		
			工学的アプローチについて		
2	コンピュータの概念と基本的な構造を説明する。	CMP.cf.5	コンピュータの構成、バイトとビット、アドレス	CMP.cf.5	コンピュータの構成、バイトとビット、アドレスを説明できる
		CMP.cf.5	メモリ、CPU、I/O	CMP.cf.5	メモリ、CPU、I/Oを説明できる
		CMP.cf.5	ノイマン型コンピュータの仕組み	CMP.cf.5	ノイマン型コンピュータを説明できる
3	コンピュータの命令語の構成と種類、割り込みについて説明する。	CMP.cf.5	アセンブリ言語の命令の構成、基本的な命令の種類、割り込み	CMP.cf.5	アセンブリ言語の命令の構成、基本的な命令の種類、割り込みを説明できる
4	2進数とN進数、数の計算機上での表現、各種文字コード、データの表現方法について説明する。	CMP.cf.5	2進数の演算、N進数、2進数と8進数、10進数、16進数の相互変換	CMP.cf.5	2進数の演算ができる。N進数の演算ができる。2進数とN進数の相互変換ができる
		CMP.cf.5	BCD、EBCDIC、ASCII、漢字コード	CMP.cf.5	コードの意義について説明できる
		CMP.cf.5	計算機上での各種データの表現	CMP.cf.5	計算機上でのデータの表現方法について説明できる
5	メモリ階層について紹介し、仮想記憶方式とキャッシュの仕組みについて説明する。	CMP.cf.5	仮想記憶方式の仕組み	CMP.cf.5	仮想記憶方式について概略を説明できる
		CMP.cf.5	キャッシュの仕組み	CMP.cf.5	キャッシュについて概略を説明できる
6	入出力、割込み、インタフェース、ネットワーク技術とネットワーク階層について説明する。	CMP.cf.5	入出力、割込み、インタフェース	CMP.cf.5	入出力データ、機器、割り込み、インタフェースについて説明できる
		CMP.cf.5	ネットワーク技術、ネットワーク階層、	CMP.cf.5	ネットワークの仕組み、ネットワーク技術について概略を説明できる
7	プログラムの概念と定義、特長、言語処理系、各種のプログラム言語について説明する。	CMP.cf.5	プログラムの概念と定義、利点	CMP.cf.1	プログラムとは何か説明できる
		CMP.cf.5	コンパイラ、インタプリタ	CMP.cf.10	言語処理系について概要を説明できる
		CMP.cf.5	手続き型、オブジェクト指向型、関数型、論理型など各種プログラム言語	CMP.cf.9	プログラム言語について説明できる
8	プログラムの基本となる制御構造、アルゴリズムの記法、構造化プログラムについて説明する。	CMP.cf.5	順接、選択、繰り返し	CMP.cf.1	プログラムの制御構造を説明できる
		CMP.cf.5	アルゴリズムとその記法	CMP.cf.2	アルゴリズムとその記法について説明できる
9	構造化プログラム、プログラムとソフトウェアについて説明する。	CMP.cf.5	構造化プログラムの概念と利点、モジュール特性とメトリクス	CMP.cf.5	構造化プログラムの概念、利点について説明できる
			プログラムとソフトウェアの共通点と差異	CMP.cf.5	プログラムとソフトウェアの差異について説明できる
10	ソフトウェア工学の背景、定義、成果について説明する。		ソフトウェア工学の背景(歴史的経緯、実例、工学の必要性)		ソフトウェア工学の定義と必要性を、事例を交えて理解している。
			ソフトウェア工学の定義(プロダクト、プロセス、ピープル、プロジェクト、QCD)		ソフトウェア工学を構成する4つの活動: プロダクト、プロセス、プロジェクト、ピープル(人間系)の意味と相互の関係および必要性を理解している。
11	ソフトウェア開発プロセスとプロセスモデルについて説明する。		プロダクトの要求、分析、設計、構築、V&V、保守・再利用		プロダクトの開発・保守プロセスにおける要求、分析、設計、構築、V&V、保守・再利用の位置づけと必要性を理解している。
			ソフトウェアライフサイクルプロセス、プロセスモデル		ソフトウェアライフサイクルプロセスの意味と必要性、主要なプロセスの種類・位置づけ、および、主要なプロセスモデル(ウォーターフォール、イテラティブ、インクリメンタル、アジャイル)を理解している。
12	開発プロジェクトとプロジェクト管理、要員、ソフトウェア品質、コスト、納期について説明する。		プロジェクト		ソフトウェア開発および工学活動におけるプロジェクトの定義とプロジェクトマネジメントの意味・必要性を理解している。
			ピープル(人間系)		ソフトウェア開発および工学活動における人間系の位置づけと重要性、および、主要な利害関係者の種類と役割・プロダクトやプロセス・プロジェクトとの関係を理解している。
			品質・コスト・納期(QCD)		ソフトウェア開発および工学活動における品質、コスト、納期の意味と重要性、主要な基準、および、それらを達成するための主要な技術との関係を理解している。

授業展開	学習成果	トピック	学習目標	発展学習目標
13	システムエンジニアリングと各種システムについて説明する。	CMP.cf.6 システムとは、システム概念と利点、システムの分類	CMP.cf.6 システムとは何か、どういったシステムがあるかを説明できる	
		CMP.cf.6 情報システムとコンピュータシステムの違い	CMP.cf.6 情報システムとコンピュータシステムを説明できる	
		CMP.cf.6 システムとハードウェアとソフトウェアの関係	CMP.cf.6 システムとハードウェア、ソフトウェアを説明できる	
		CMP.cf.6 組み込みシステムの紹介	CMP.cf.6 組み込みシステムを説明できる	MAA.tm.7 組み込みシステムをモデリングできる
		CMP.cf.6 企業情報システムの紹介	CMP.cf.6 企業情報システムを説明できる	MAA.tm.6 企業情報システムをモデリングできる
14	重要事項のまとめ			
15	期末試験			
教科書・参考書	ロジャー・プレスマン: 実践ソフトウェアエンジニアリング, 日科技連, 2005. エリック・プロディ: ソフトウェアエンジニアリング, 翔泳社, 2004			

授業科目名	確率統計				
単位数	2				
開設学期	2年生前期				
目的	確率モデルや統計分析手法の実装を必要とするソフトウェアの開発に必要な、およびソフトウェア開発プロジェクトを通じて測定されたデータの分析に必要な確率統計の基礎知識の習得を目的とする。				
概要	科学技術分野などのコンピュータシステムでは、確率モデルや統計分析手法のソフトウェアによる実現が求められる場合が少なくない。また、ソフトウェア工学の観点からは、ソフトウェア開発プロジェクトを定量的に管理することが、品質・コスト・納期の問題解消に貢献する。したがって、ソフトウェア開発およびその管理の両面において、確率統計の基礎知識は必須である。以上の問題意識に基づいて、本講義では、確率統計および統計解析の基礎知識を学ぶ。				
目標	確率統計の基礎知識を理解し、事象の確率計算、データの適切な表現、回帰分析や検定など基本的な統計解析が行えることを目標とする。				
先修科目					
関連科目					
授業方法					
評価方法・基準	中間試験や期末試験、レポートにより評価する。				
授業展開		学習成果	トピック	学習目標	発展学習目標
1	授業の概要と目的および進め方について説明する(※講義0.3h)。記述統計と推測統計の違い、母集団と標本の意味、変数の種類、一変数データのグラフ表現における留意点について説明する。	FND.ef.2	記述統計と推測統計、標本と母集団、データの種類(一変数データ、二変数データ)、量的変数と質的変数、連続変数と離散変数、各種グラフ、ヒストグラム、分布の見方(対称性、単峰、外れ値)	記述統計と推測統計の違いを説明できる。 母集団から標本をとる意義について説明できる。 変数の種類を理解し、質的変数には四則演算が適用できないといった変数の性質・制約を説明できる。 データを適切なグラフで表現できる。 ヒストグラムの形状からデータの分布の特徴を説明できる。例えば、「右(または左)に偏っている」など。	
2	一変数データの中心とばらつきを記述する基本的な統計量と、データの分布に用いられる箱ひげ図について説明する。	FND.ef.2	母平均、標本平均、メジアン、範囲、母分散、標本分散、母標準偏差、標本標準偏差、標準化得点、パーセント点、四分位点、五数要約、箱ひげ図	母数と統計量の違いを説明できる。 平均や標準偏差などの統計量を計算により求めることができる。 平均とメジアンの違いを理解し、どのような分布のときにメジアンを用いるのが望ましいのか説明できる。 分散および標準偏差の式の構造を理解し、平均とデータの差を二乗する理由について説明できる。また、標準偏差は元データと同じ単位を持つ統計量であることを説明できる。 データの標準化得点を計算し、平均からの離れ具合を理解できる。 データを五数要約で表し、外れ値を考慮した箱ひげ図で表すことができる。	分散や標準偏差の計算において、なぜ絶対値を用いずに二乗を用いているのか説明できる。 尖度や歪度を学び、それらの統計量を計算により求めることができる。 標本分散を求めるのになぜn-1で除算するのか、その理由について直感的な説明ができる。
	有効数字の計算における留意点を学ぶ。	FND.mf.9	有効数字	統計量の計算にあたり、有効数字を考慮した計算ができる。	
3	二変数データを記述する方法として、相関と散布図について説明する。また、最小二乗法により回帰直線を得る方法を説明し、モデル式を用いた予測が行えることを説明する。	FND.ef.2	散布図、クラスターと外れ値、共分散、相関係数、最小二乗法、回帰直線	散布図を作成し、分布の特徴を説明できる。 相関係数の式の構造を理解できる。 相関係数の数値の範囲、数値が表す意味について説明できる。 相関係数が同一でも散布図の上では異なる分布のパターンがあることを理解する(アンスコムの数値例)。 最小二乗法によりなぜ回帰直線が得られるのか理解できる。 得られた回帰直線を用いて、所与のデータを用いて予測できる。	最小二乗法により回帰直線が得られることを、偏微分により照明できる。
4	確率の基本概念について説明する。	FND.mf.6	単純事象、事象、相互排他的、標本空間、ベン図、確率の定義、順列、組み合わせ、和事象、積事象、余事象、条件付き確率、独立事象、和の法則、積の法則	単純事象を数え上げることで、所与の事象の確率を求めることができる。 和の法則および積の法則を用いて、確率を求めることができる。 条件付き確率を用いた確率の計算ができる。	
5	離散確率変数と連続確率変数の違いを説明し、離散確率変数の確率分布について説明する。	FND.mf.6	ベイズの定理、確率変数、確率分布、離散確率変数と連続確率変数、離散確率変数の期待値(確率変数の平均)、離散確率変数の分散と標準偏差	離散確率変数と連続確率変数の違いについて説明できる。 離散確率変数の期待値(平均)、分散、標準偏差を計算できる。	
6	二項分布について説明し、二項確率を数式により求める方法と、累積二項確率分布表から求める方法を説明する。	FND.mf.6	二項確率変数の性質、二項確率の計算(数式による/累積二項確率分布表による)、二項確率変数の平均と分散	所与の状況において二項確率変数として考慮してよいか否かの判断ができる。 二項確率を数式により計算できる。 二項確率を累積二項確率分布表から求めることができる。 二項確率変数の平均と分散を計算できる。	ポアソン分布など、その他の分布について学ぶ。
7	正規分布について説明し、正規分布に従ったデータを標準化して正規分布表から確率を求める方法を説明する。	FND.ef.2	正規分布、標準正規分布、標準化、正規分布表、確率の計算(正規分布表による)	正規分布は $\mu$ と $\sigma$ をパラメータに持っており、これらの数値によって形状が決まることを理解する。 正規分布に従ったデータを標準化し、正規分布表を調べることで確率を計算できる。	二項確率変数を正規分布に近似させて確率を求める方法を学ぶ。
	測定値と誤差の関係を説明し、多くの測定値が正規分布に従うことを説明する。	FND.mf.9	測定値と誤差	測定値と誤差の関係を理解し、多くの測定値が正規分布に従うことを理解する。	
8	サンプリングの方法を説明し、母集団分布と標本分布の違いについて説明する。また、中心極限定理について説明する。	FND.ef.2	単純ランダムサンプリング、その他のランダムサンプリング手法、母集団分布と標本分布の違い、標準誤差、中心極限定理	サンプリング方法には複数の方法があることを理解する。 母集団分布と標本分布の違いについて説明できる。 統計量の標本分布における標準偏差のことを標準誤差と呼ぶことを理解する。 中心極限定理により、母集団分布が任意の形状であっても、サンプルサイズが十分大きければ標本分布を正規分布に近似させられることを理解する。	
9	標本平均および標本比率の標本分布における平均と標準誤差について説明する。また、統計的工程管理(SPC)の基礎について説明する。	FND.ef.2	標本平均の標本分布、標本比率の標本分布、統計的工程管理、管理図、 $\bar{x}$ -R管理図、 $p$ 管理図	標本平均および標本比率の標本分布について、それぞれの平均と標準誤差を計算できる。 統計的工程管理の基本的な考え方と見方を理解し、 $\bar{x}$ -R管理図および $p$ 管理図を作成できる。	$\bar{x}$ -R管理図、 $p$ 管理図以外の管理図について、それぞれの特徴や使い方が理解できる。
10	点推定と区間推定について説明し、許容誤差を与えることで必要なサンプルサイズを決定できることを説明する。	FND.ef.2	点推定と区間推定、不偏性、不偏推定量、許容誤差、信頼係数、母平均の点推定と区間推定、母比率の点推定と区間推定、サンプルサイズの決定	推定における不偏性と分散性の違いについて説明できる。 信頼係数、サンプルサイズ、母標順偏差によって許容誤差が決まることを理解できる。 母平均と母比率について、それぞれの点推定と区間推定を行える。 区間推定における信頼係数の意味について理解できる。	標本分散が不偏推定量であることを、数式を用いて証明できる。

授業展開	学習成果	トピック	学習目標	発展学習目標
11	統計的仮説検定の基本的な考え方を説明し、大標本における統計的仮説検定(母分散が既知の場合)の方法を説明する。	FND.ef.2 検定の手順、帰無仮説と対立仮説、片側検定と両側検定、検定統計量、p値、棄却域、検定における結論付けのしかた、第一種の過誤、第二種の過誤、母平均の検定	検定の手順について説明できる。 適切な帰無仮説および対立仮説を立てることができる。 母分散が既知の場合における、母平均に関する検定(z検定)が行える。 検定において生じる二種類の過誤について理解し、第一種の過誤が起きる確率が有意水準 $\alpha$ と同一であることを理解する。 帰無仮説が棄却されない場合に、「帰無仮説を棄却できるとはいえない」という表現で結論づけることの妥当性を理解する。	第二種の過誤と検出力の関係について説明できる。
12	大標本における統計的仮説検定について、二群の母平均の差の検定と、母比率の検定について説明する。また、信頼区間と検定の関係性について説明する。	FND.ef.2 二群の母平均の差の検定、母比率の検定、信頼区間と検定	二群の母平均の差の検定が行える。 母比率の検定が行える。 検定において帰無仮説が棄却される場合は、信頼区間が母平均を含んでいない場合と等しいことを理解する。	
13	小標本における統計的仮説検定について、検定の対象によって異なる分布を用いることを説明する。	FND.ef.2 t分布、F分布、カイ二乗分布、自由度、母平均の検定、母平均の差の検定(等分散の場合/分散が異なる場合)、対応のある二群の母平均の差の検定、母分散の検定、二群の母分散の比の検定	t分布、F分布、カイ二乗分布の違いを理解できる。 検定の対象によって、異なる検定統計量、異なる分布を用いることを理解できる。	
14	重要事項のまとめ			
15	期末試験			
教科書・参考書	Mendenhall, W., Beaver, R.J. and Beaver, B.M., Introduction to Probability and Statistics, 12th ed., Thomson, 2006. 東京大学教養学部統計学教室(編), 統計学入門, 東京大学出版会, 1991.			
備考				

授業科目名		離散数学					
単位数		2					
開設学期		2年生後期					
目的		ソフトウェア工学において必要になる離散数学の様々な分野について学習することを目的とする。					
概要		グラフ理論、代数学、数論、数え上げの基礎について学ぶ。					
目標		ソフトウェア工学の理解に必要な離散数学の知識を獲得する。					
先修科目							
関連科目							
授業方法		座学					
評価方法・基準		中間及び期末試験の結果により評価を行う					
授業展開		学習成果	トピック	学習目標	発展学習目標		
1	授業の概要と目的および進め方について説明する(※講義0.3h)。理解に必要な、集合論の基礎を説明してから、基本的な諸概念を、実例などを交えて説明をする。	FND.mf.5	グラフの基本的な諸概念の定義(頂点と辺、次数、木、サイクル、連結性、同型)	FND.mf.5	グラフの基本的な諸概念について理解をする。	FND.mf.5	
2	様々なグラフの中から重要な応用(例えば、オイラー路)について教える。	FND.mf.5	オイラーグラフ、ハミルトングラフ	FND.mf.5	グラフの具体的な例を学び、計算機科学における応用について理解をする。	FND.mf.5	
3	グラフの探索アルゴリズムとその実装について説明する。	FND.mf.5	木グラフ、探索アルゴリズム(DFS, BFS)	FND.mf.5	グラフの表現方法とともに、グラフの探索アルゴリズムについて理解をする。	FND.mf.5	
4	群に関する基礎的な諸概念を説明する。	FND.mf.11	群の公理、部分群、準同型写像、同型定理	FND.mf.11	群の公理系に関する基本的な諸概念と、基本的な定理について理解をする。	FND.mf.11	より高度な群論の概念、定理(半群、変換モノイド、巡回モノイド)について理解でき、代数構造に対する理解を深めることが出来る。
5	環に関する代数的基礎について説明をする。	FND.mf.11	公理、イデアル、剰余環、準同型写像、直和、直積	FND.mf.11	環の公理系に関する基本的な諸概念と、基本的な定理について理解をする。	FND.mf.11	
6	体に関する代数的基礎について説明をする。	FND.mf.11	公理、代数的拡張、代数的閉包	FND.mf.11	体の公理系に関する基本的な諸概念と、基本的な定理について理解をする。	FND.mf.11	
7	計算機科学において広く用いられている、順序集合とそれ上の代数構造である束について学び、ブール代数まで拡張した代数系の説明を行う。	FND.mf.11	順序集合と束、束の代数的構造、分配束、ブール代数(分配可補束)	FND.mf.11	束とブール代数の公理系に関する基本的な諸概念と、基本的な定理について理解をする。	FND.mf.11	命題論理との関連(リンデンバウム代数)、表現定理について学ぶことで、ブール代数と論理体系との関わり合いについて理解を深めることが出来る。
8	数(自然数、整数、素数)に関する基礎知識を説明する。	FND.mf.10	数の定義、素因数分解、剰余算、素数の判定、ユークリッドの互除法	FND.mf.10	整数を中心とした数に関する基礎理論を理解する。	FND.mf.10	
9	整数に関する基本的定理について説明を行う。	FND.mf.10	中国の剰余定理、Wilsonの定理、フェルマーの小定理	FND.mf.10	整数論の基礎について理解をする。	FND.mf.10	
10	整数論の暗号への応用について説明を行う。	FND.mf.10	オイラーの公式、暗号概論	FND.mf.10	整数論と暗号についての、暗号の基礎(アフィン暗号、公開鍵暗号、RSA暗号)にどのように整数論が用いられるかを理解する。	FND.mf.10	
11	数え上げの基礎概念について説明をする。	FND.mf.4	順列、組み合わせ、重複順列、重複組み合わせ	FND.mf.4	順列と組み合わせの基本的な諸概念について理解をする。	FND.mf.4	
12	順列と組み合わせ理論についてより複雑な概念といくつかの結果について説明を行う。	FND.mf.4	二項定理、多項係数、漸化式	FND.mf.4	順列と組み合わせに対するより複雑な諸概念について理解をする。	FND.mf.4	
13	組み合わせ理論において重要なカタラン数、スターリング数、フィボナッチ数などについて説明をする。	FND.mf.4	カタラン数、スターリング数、フィボナッチ数	FND.mf.4	組み合わせ理論に現れる数の性質、表現とその事例を基に、組み合わせ問題について理解を深める。	FND.mf.4	
14	重要事項のまとめ						
15	期末試験						
教科書・参考書	やさしい組み合わせ数学、西岡弘明、コロナ社(1999)、グラフ・ネットワーク・組み合わせ論、藤重悟、共立出版(2002)						
備考							

授業科目名	プログラミング基礎					
単位数	2					
開設学期	2年生後期					
目的	プログラミング言語によりソフトウェアをどのように作成するかを理解し、そのうえで効率よく問題を解決するためのアルゴリズムについて基礎を学ぶ					
概要	C言語を対象にして、プログラミング言語によりソフトウェアをどのように作成するのかという基本を理解する。さらに、問題を解決するためのアルゴリズムの概念と実装の基礎について理解する					
目標	プログラムの作成方法や実行形式への変換方法を学ぶ。変数、データ型、演算、分岐、繰り返し、配列操作などの基本的なプログラムの文法を理解し、これらを用いてプログラムが作成できるようになる。 計算量の概念を身につけ、効率の良いプログラムを作成するためのアルゴリズムやデータ構造の存在を知る					
先修科目						
関連科目						
授業方法	講義、机上演習  ・プログラミング基礎については机上演習を行う。					
評価方法・基準	下記により評価する。 ・机上演習による提出物 ・中間および期末試験の結果					
授業展開	学習成果	トピック	学習目標	発展学習目標		
1	講義を通して学ぶプログラミング言語であるC言語の概要及び発展の歴史を学び、さらにプログラミング環境の基本的な使い方を習得する	CMP.cf.9 CMP.cf.9 CMP.cf.9 CMP.cf.9	トピック プログラミング言語の発展の歴史及び言語としての特徴を学ぶ アルゴリズムとプログラミング言語の関係の概要を学ぶ ソースファイルやバイナリファイルなどの各種ファイルとコンパイラ等の基本的な開発ツールを学ぶ 単純な入出力を行うプログラムを題材にプログラムの作成、コンパイル、実行までの手順を学ぶ	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	開発環境 単純な入出力	
2	変数の宣言方法、データ型、演算子などを学び、それらを使ってプログラムを作成する	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	変数の宣言方法、基本データ型、参照データ型、数値リテラル、文字リテラルを学ぶ 代入演算子、算術演算子、関係演算子などとそれらの優先順位を学ぶ 文、スコープ、ブロック、コメントなどのプログラムの構成要素を学ぶ 変数や演算子を使ったプログラミング課題を行う(または、穴埋め問題を行ったり、プログラム例を解説する)	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	変数、型、式、代入	
3	分岐と繰り返しの2種類の制御文を学ぶ	CMP.cf.1 CMP.cf.1 CMP.cf.1	if文やswitch文などの分岐の定義方法を学ぶ forループ、whileループ、do ~ whileループなどの繰り返しの定義方法を学ぶ 分岐や繰り返しを使ったプログラミングを行う(または、穴埋め問題を行ったり、プログラム例を解説する)	CMP.cf.1 CMP.cf.1 CMP.cf.1	条件判定と繰り返しの制御構造	
4	1次元配列や多次元配列の操作方法を学ぶ	CMP.cf.1 CMP.cf.1	配列の初期化、配列の要素に対する繰り返し処理などを学ぶ 配列を使ったプログラミング課題を行う(または、穴埋め問題を行ったり、プログラム例を解説する)	CMP.cf.1 CMP.cf.1	配列、レコード	
5	関数の定義方法を学ぶ	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	関数を定義する意義を学ぶ 関数とその引数や戻り値の宣言方法を学び、さらに値引数を実引数に影響を与えないことを学ぶ プロトタイプ宣言とヘッダーファイル、さらにinclude文でヘッダーファイルを参照できることを学ぶ 関数についての簡単なプログラミング課題を行う(あるいは、穴埋め問題を行ったり、プログラム例を解説する)	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	関数と引数受け渡し	CMP.cf.1 構造的分解
6	文字列操作、ポインター、再起呼び出しを学ぶ	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	文字列のデータ構造を学ぶ ポインターの概念を学び、ポインターを使った文字列の要素の辿り方を学ぶ 関数の引数にポインターを用いることを学び、その場合の実引数への影響を理解する 再帰呼び出しの概念を学び、さらに再帰呼び出しを用いる簡単なアルゴリズムの例を学ぶ ポインターを引数とする関数のプログラミング課題を行う(あるいは、穴埋め問題を行ったり、プログラム例を解説する)	CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1 CMP.cf.1	文字列と文字列処理 ポインターと参照 再帰の概念と実装	CMP.cf.1 メモリ内でのデータ表現

授業展開	学習成果	トピック	学習目標	発展学習目標			
7	イベント駆動型プログラミングとプログラムの品質や信頼性を高める方法を学ぶ	CMP.cf.1	GUI操作で発生するイベントに基づくプログラムの動作方法について学ぶ	CMP.cf.1	イベント処理方法	CMP.ct.2	コードの再利用とライブラリ
		CMP.cf.1	例外やエラーの通知方法やハンドリングするためのプログラミング方法や戦略について学ぶ	CMP.cf.1	エラーハンドリング、例外処理	CMP.ct.4	パラメータ化と汎化
		CMP.cf.1	アサーションや契約による設計(DbC)などにより、プログラムの信頼性を高める方法を学ぶ	CMP.cf.1	アサーション、契約による設計(DbC)	CMP.ct.5	防御的プログラミング
		CMP.cf.4	カプセル化や階層化により、プログラムの抽象化を行うことを基本的なデータ構造を例にして学ぶ	CMP.cf.4	抽象化(カプセル化や階層化など)、基本的なデータ構造(スタック、キュー、ハッシュ)	CMP.ct.6	フォールトトレラント
8	プログラミング環境とツールを学ぶ	CMP.tl.1	ソースコードの編集、ビルド、デバッグなど一般的な統合開発環境で提供されている機能を学ぶ	CMP.tl.1	開発環境	CMP.cf.1	デバッグ戦略
		CMP.tl.2	画面レイアウトの定義など一般的な統合開発環境で提供されているGUI構築ツールを学ぶ	CMP.tl.2	GUI構築ツール	CMP.tl.4	アプリケーション指向言語(スクリプト言語、ビジュアル言語、ドメイン特化言語、マークアップ言語、マクロなど)
		CMP.tl.3	単体テストの概念を学び、単体テストを自動化するツールを使ったテストコードの書き方を学ぶ	CMP.tl.3	単体テストツール		
		CMP.tl.5	プロファイリングやメモリークックの検出ツールなどについて学ぶ	CMP.tl.5	プロファイリングツール		
9	アルゴリズムの基礎	CMP.cf.1, CMP.cf.2	アルゴリズムと問題解決	CMP.cf.1, CMP.cf.2	問題解決戦略という考え方を身につけ、問題解決過程におけるアルゴリズムの役割を理解できる		
		CMP.cf.1, CMP.cf.2	アルゴリズムにおける計算量の概念、計算量の上界と漸近的計算量	CMP.cf.1, CMP.cf.2	計算量の概念、計算量の上界と漸近的計算量について理解できる。最良/平均/最悪の区別について説明できる O記法, o記法, Ω記法, Θ記法, ω記法		
		CMP.cf.1, CMP.cf.2	アルゴリズムの実現戦略	CMP.cf.1, CMP.cf.2	時間と領域のトレードオフについて理解し、アルゴリズムの実現戦略を検討できる		
10	アルゴリズム設計手法	CMP.cf.2	カズクアルゴリズム、どん欲アルゴリズム	CMP.cf.1, CMP.cf.2	カズクアルゴリズム、どん欲アルゴリズムの手法について理解する ヒューリスティクスの概念を理解する		
		CMP.cf.1, CMP.cf.2	再帰的アルゴリズム	CMP.cf.1, CMP.cf.2	再帰の概念について復習する 分割統治法、バックトラック法、分枝限定法などの手法を理解する		再帰的アルゴリズムを実際にプログラムとして記述する
		CMP.cf.2	パターン照合アルゴリズムと近似数値アルゴリズム	CMP.cf.2	パターン照合アルゴリズムや近似数値アルゴリズムについて代表的な例に触れる		
11	基礎的なアルゴリズム	CMP.cf.2	単純な数値計算アルゴリズム	CMP.cf.2	単純な数値計算アルゴリズムを学ぶ		
		CMP.cf.2	整列アルゴリズムO(N^2)~選択法、挿入法	CMP.cf.2	整列アルゴリズムO(N^2)~選択法、挿入法について理解する		
		CMP.cf.2	整列アルゴリズムO(NlogN)~クイックソート、ヒープソート、マージソート	CMP.cf.2	整列アルゴリズムO(NlogN)~クイックソート、ヒープソート、マージソートについて理解する	CMP.cf.2	いくつかのソートアルゴリズムを実際にプログラムとして記述/実行し、その効率について体験する
12	探索アルゴリズムとデータ構造	CMP.cf.1, CMP.cf.2	基本的なデータ構造と探索アルゴリズム(逐次探索、二分探索、ハッシュ)	CMP.cf.1, CMP.cf.2	探索アルゴリズム(逐次探索、二分探索、ハッシュ)について理解する 配列、連結リスト、二分探索木、スタック、キューの概念と実装法を理解する	CMP.cf.1, CMP.cf.2	基本的リスト構造とその上での探索を実際にプログラムとして記述する
		CMP.cf.2	グラフ構造と探索アルゴリズム	CMP.cf.2	グラフ表現(隣接リスト、隣接行列)のデータ構造を理解する グラフ探索における深さ優先、幅優先の概念を理解する 最短経路アルゴリズム(ダイクストラ、フロイド)を理解する		
13	分散アルゴリズムと計算可能性	CMP.cf.2	分散アルゴリズム	CMP.cf.2	分散アルゴリズムにおける合意形成とリーダー選出、終了判定などの操作について理解する。	CMP.cf.2	推移閉包、最小全域木、トポロジカルソートの概念に触れる 経路探索アルゴリズムを実際にプログラムとして記述する
		CMP.cf.2	計算機械の基礎	CMP.cf.2	フォールトトレランス、安定性と言った概念をしる 有限状態機械、文脈自由言語の概念を理解する		
		CMP.cf.2	計算可能性	CMP.cf.2	実質的に計算可能な問題と不可能な問題が存在することを知り、計算不可能関数について理解する。停止問題、計算不可能性から導かれることについて理解する		
14	重要事項のまとめ						
15	期末試験						
教科書・参考	C言語の場合: 柴田望洋, 明解C言語入門編, ソフトバンククリエイティブ, 2004 浅野, 増沢 利光, 和田 幸一, アルゴリズム論 (IT Text) (単行本), オーム社, 2003 計算とアルゴリズム(新コンピュータサイエンス講座): 浅野孝夫, 今井弘, オーム社, 2000						
備考	本カリキュラムは、プログラミング初心者を対象とした弊社の新人研修の講義及び実習内容と、柴田望洋さんの「新版明解C言語入門編」を参考に作成しました						

授業科目名	論理と計算理論	
単位数	2	
開設学期	2年前期	
目的	本講義では、論理学と計算理論の基礎を学ぶことを目的とする。	
概要	ソフトウェアを形式的に特徴づけたり解析するためには、計算や論理に関する知識が不可欠である。本講義では、論理学や計算理論を中心に、関連する議論も含めて学習する。	
目標	<ul style="list-style-type: none"> <li>集合、関係、関数に関する基本的な知識を習得する。</li> <li>命題論理および述語論理に関する基本的な知識を習得する。</li> <li>言語理論と計算理論に関する基本的な知識を習得する。</li> </ul>	
先修科目		
関連科目	離散数学、データ構造とアルゴリズム	
授業方法	講義と演習	
評価方法・基準	期末試験、または、必要に応じて中間試験の結果により評価を行う。	
授業展開	学習成果	トピック

学習成果	トピック	学習目標	発展学習目標
FND.mf.1	内包的表記、外延的表記、集合演算	FND.mf.1 集合の表現法と意味、および、基本的な演算について理解する。	FND.mf.1 集合を用いて証明を行うことができるようになる。集合を用いて理論を構築できるようになる。
FND.mf.1	有限集合、無限集合、濃度	FND.mf.1 集合の諸概念(有限集合、無限集合、濃度など)について理解する。	FND.mf.1 集合を用いて証明を行うことができるようになる。集合を用いて理論を構築できるようになる。
FND.mf.1	関係、写像、半順序、全順序、関数、単射、全射、全単射、部分関数、全関数	FND.mf.1 集合に基づいた関係とその分類について説明する。関数、および、その分類について理解する。	FND.mf.1 関数や関係を用いて証明を行うことができるようになる。関数や関係を用いて理論を構築できるようになる。
FND.mf.2	命題論理式、真理値表、トートロジー、付値、充足可能性、同値性、標準形	FND.mf.2 命題論理の論理式と諸概念(真理値表、トートロジー、同値性、標準形など)について理解する。	FND.mf.2 より発展した論理を理解する(様相論理、直観主義論理など)。充足可能性問題解決器(SATソルバ)について理解する。
FND.mf.3	形式的体系、証明図、健全性、完全性	FND.mf.3 命題論理の形式的体系とそれによる証明について説明する。形式的体系の性質について理解する。	FND.mf.3 様々な形式的体系(シーケント計算、自然演繹など)を理解する。より発展した論理(様相論理、直観主義論理など)の形式的体系を理解する。形式的体系と他理論との関係を理解する(λ計算など)。
FND.mf.2	述語論理式、述語変数、量化記号、束縛変数、自由変数	FND.mf.2 述語論理の論理式と諸概念(述語変数、量化記号、束縛変数、自由変数など)について理解する。	FND.mf.2 より発展した論理を理解する(様相論理、直観主義論理など)。証明支援システム、自動定理証明システムについて理解する。
FND.mf.2	構造、解釈	FND.mf.2 構造と述語論理の意味について理解する。	FND.mf.2 より発展した論理を理解する(様相論理、直観主義論理など)。証明支援システム、自動定理証明システムについて理解する。
FND.mf.2	冠頭標準形、恒真性、充足可能性	FND.mf.2 述語論理の諸概念(冠頭標準形、恒真性、充足可能性など)について理解する。	FND.mf.2 より発展した論理を理解する(様相論理、直観主義論理など)。証明支援システム、自動定理証明システムについて理解する。
FND.mf.3	形式的体系(シーケント計算、自然演繹など)、健全性、完全性	FND.mf.3 述語論理の形式的体系とそれによる証明について理解する。形式的体系の性質について理解する。より発展した論理があることを理解する。	FND.mf.3 様々な形式的体系(シーケント計算、自然演繹など)を理解する。より発展した論理(様相論理、直観主義論理など)の形式的体系を理解する。形式的体系と他理論との関係を理解する(カリー-ハワード対応など)。モデル検査について理解する。
FND.mf.7	有限オートマトン、言語、	FND.mf.7 有限オートマトンの定義と、その表現能力について理解する。	FND.mf.7 より発展したオートマトン(プッシュダウンオートマトン、木オートマトン、チューリング機械など)を理解する。
FND.mf.7	決定性オートマトン、非決定性オートマトン、等価性、計算量	FND.mf.7 決定性オートマトン、非決定性オートマトンの定義と諸概念(等価性など)について理解する。計算量(P, NP, PSPACE)について理解する。	FND.mf.7 より発展したオートマトン(プッシュダウンオートマトン、木オートマトン、チューリング機械など)を理解する。
FND.mf.7	正規表現、接続、閉包	FND.mf.7 正規表現の定義と諸概念(有限オートマトンとの等価性など)について理解する。	FND.mf.7 より発展したオートマトン(プッシュダウンオートマトン、木オートマトン、チューリング機械など)を理解する。
FND.mf.8	文脈自由文法、導出、導出木	FND.mf.8 文脈自由文法と諸概念について理解する。	FND.mf.8 標準形などの発展した理論を理解する。文脈自由文法の分類と構文解析手法を理解する(LL, LR, LALRなど)。

14	重要事項のまとめ	
15	期末試験	
教科書・参考書	田辺誠・中島玲二・長谷川真人,「コンピュータサイエンス入門」, 岩波書店 小野寛晰, 「情報科学における論理」, 日本評論社 J. ホップクロフト・J. ウルマン・R. モトワニ(野崎・町田・高橋・山崎訳)「オートマトン 言語理論 計算論(第2版)」, サイエンス社	
備考		



授業科目名	オペレーティングシステム基礎・データベース基礎
単位数	1
開設学期	2年生前期
目的	オペレーティングシステムの基本的な役割、データベースのモデル、言語、設計手法、運用管理手法を学習することを目的とする
概要	オペレーティングシステムは、プログラマがハードウェアを制御できるよう抽象化し、ユーザの間で共有する資源を管理する。このエリアのトピックは、現在のオペレーティングシステムの設計に影響を与える課題について、取り扱う。 また、大量のデータを効率よく利用するために欠かすことのできないデータベースに関して取り上げ、データベースのモデル、言語、設計手法、運用管理手法を学習する。
目標	オペレーティングシステムの役割と基本的概念を把握し、コンピュータシステムの働きを理解するとともに、各種基本構成要素とその働きについて理解することを目的とする。 データベースの歴史から組織での必要性を理解し、データベース言語SQLが使用できること、論理データモデルのモデル化方法と物理データベース設計方法を用いてデータベース設計ができることを目標とする。また、データベース管理システムに関しての知識を持ち、運用管理の方法を理解していることを目標とする
先修科目	
関連科目	
授業方法	講義、机上演習
評価方法・基準	下記により評価する。 ・机上演習による提出物 ・中間および期末試験の結果
授業展開	

1	学習成果	トピック	学習目標	発展学習目標
1	授業の概要と目的および進め方について説明する(※講義0.3h)。 オペレーティングシステム概念や位置づけを説明する。	オペレーティングシステムの役割と目的	現在のオペレーティングシステムについて、その役割や目的、機能を理解し、説明できる。	
		オペレーティングシステムの歴史	原始バッチシステムから洗練されたマルチユーザーシステムまで、オペレーティングシステムがどのように進化したかを理解する。	オペレーティングシステム設計に固有のトレードオフを理解できる。
		設計課題(効率性、頑健性、柔軟性、移植性、信頼性、セキュリティ、互換性) セキュリティ、ネットワーク、マルチメディア、Windowsの影響 運用と管理	潜在的脅威を防御するセキュリティ設計とオペレーティングシステムとの関係について理解する。 ブートストラップとローダ、シャットダウン、各種ログについて、その役割を理解し、説明できる。	オープンソースソフトウェアやインターネット人口増加のような課題が、オペレーティングシステムの設計に影響を与えていることを理解し、説明できる。
2	オペレーティングシステムの仕組みについて説明する。	構成方法(モノリシック、階層化、モジュール化、マイクロカーネル)	論理的な階層の概念を理解し、説明できる。	オブジェクト指向、モジュール化、マイクロカーネル、階層化のようなオペレーティングシステムのさまざまな構成について、それらの特徴を比較し、論じることができる。
		抽象化、プロセス、資源	階層化の利点を理解し、説明できる。	
		ユーザモードからカーネルモードへの移行や保護のための概念	計算リソースを、アプリケーションソフトウェアがどのように使い、システムソフトウェアがどのように管理するのかについて理解し、説明できる。 オペレーティングシステムにおいて、ユーザモードとカーネルモードを理解し、比較することができる。	具体的なアプリケーションプログラムインターフェイス(API:Application Program Interfaces)について理解し、説明できる。
	プロセスの管理について説明する。(1)	プロセスの状態と状態遷移図	システムコールについて理解し、説明することができる。	単純な問題において、状態遷移図を描くことができる。
		構造(レディキュー、プロセスコントロールブロック(PCB)、など)	プロセスの遷移や、複数のプロセス管理のために必要な状態について理解し、説明できる。 並行性を管理するためのスタックやキューのようなデータ構造の有用性を理解し、説明できる。 PCBについて、その役割を理解し説明できる。	
3	プロセスの管理について説明する。(2)	プロセスとスレッド	プロセスとスレッドの違いを理解し、説明できる。	再入可能性(reentrant)について理解し、説明できる。
		ディスパッチとコンテキストスイッチ(文脈切り替え)	複数のプロセス操作に起因する潜在的な実行時の問題について理解し、説明できる。	
	プロセスのスケジューリングについて説明する。	横取り可能(preemptive)と横取り不可能(non-preemptive)なスケジューリング スケジューラーと方針	オペレーティングシステムでの並行性を支援するための、割込み、ディスパッチ、コンテキストスイッチを使う理由を理解し、説明できる。 FCFSや、優先度、SJF、ラウンドロビンなど、オペレーティングシステムにおけるプロセスにおける一般的なアルゴリズムを理解し、比較することができる。 ターンアラウンドタイムや、スループット、待ち時間、応答時間、CPU利用率などを理解した上で、スケジューリングの方針を論じることができる。	多重レベルスケジューリングについて理解し、説明できる。
			スケジューリングの方針と対象システムやアプリケーションとの関係を理解し、説明できる。	スケジューリングアルゴリズムに採用されている論理が、ディスクI/Oや、ネットワークスケジューリング、プロジェクトスケジューリング、コンピュータとは関係ない問題など、他の分野にも適用可能であることを確認する。
4	プロセスの同期について説明する。	並行実行の利点と欠点	オペレーティングシステムにおける並行性の必要性を理解し、説明できる。	
		相互排除とクリティカルセクション	相互排除とクリティカルセクションについて理解し、説明できる。	
		デッドロックの原因と条件と予防	デッドロックが発生する必要条件を理解し、説明できる。 デッドロックの防止、予防、検出と回復、について理解し、説明できる。	
		モデルとメカニズム(セマフォ、モニタ、状態変数、ランデブ)	並行システムの動作を理解し、オペレーティングシステムにおいて使われる各種メカニズムについて、それぞれの利点と欠点について理解し、説明できる。	生産者消費者問題や、哲学者の食事問題、読み書き問題など、並行システムのモデル化とその解法について論じることができる。
		プロセス間通信	sendとreceive、同期通信と非同期通信について理解し、説明できる。	パイプ、クライアントサーバモデル、メールボックス、ポート、遠隔手続き呼出しについて理解し、説明できる。

授業展開	学習成果	トピック	学習目標	発展学習目標
5	メモリ管理について説明する。	物理メモリとメモリ管理機構の調査 メモリ割付 オーバーレイとスワッピング メモリ保護 ページングとセグメンテーション	メモリの階層構造を理解し、メモリの大きさと速度やコストに関するトレードオフについて理解し、説明できる。 メモリ割付の各種方針について理解し、説明できる。 内部断片化と外部断片化、およびコンパクションについて理解し、説明できる。 オーバーレイとスワッピングについて理解し、説明できる。 メモリ保護について、その役割を理解し、説明できる。 ページングやセグメンテーションで使われる仮想記憶の仕組みを理解し、説明できる。	参照の局所性について理解し、説明できる。    アドレス変換テーブルのエントリについて理解し、説明できる。
6	仮想記憶について説明する。(2)  入出力制御と割込みについて説明する。	置換の方針 フェッチの方針 スラッシング デバイスの構成 割込みの方法と実現	プロセスのメモリ配置のためのさまざまな手法(FIFOやLRUなど)を理解し、それぞれの特徴を明らかにした上で説明できる。 フェッチの各種方針について理解し、説明できる。 スラッシングの問題点について理解し、説明することができる。 デバイスリストやドライバのI/Oキューの使用方法を理解し、説明できる。	ワーキングセットの概念について理解し、説明することができる。 バッファリングや、スプーリング、DMA(Direct Memory Access)について理解し、説明することができる。
7	ファイルシステムについて説明する。	レコードとブロック ディレクトリ ファイルの操作と保護 2次記憶の割付技法 バックアップ	レコードとブロックについて理解し、説明できる。 ディレクトリの操作や構成について理解し、説明できる。 ファイルの操作と保護について理解し、説明できる。 各種2次記憶の割付技法について理解し、説明できる。 バックアップの重要性を理解し、説明できる。	実際のOSに採用されている2次記憶の割付技法について理解し、説明できる。 インクリメンタルダンピング、ミラーリング、RAIDについて理解し、説明できる。
8	データベースに関連する基本的な用語や概念を説明する	CMP.cf. 11 データ、メタデータ、情報、知識、データベース、データベース管理システム リレーショナルデータモデル  リレーショナルテーブルの取り出し、格納実行機能、データベース利用要求の解釈機能に関して説明する。	データ、情報、知識、データベース、データベース管理システム、メタデータなどの重要な用語の相違を説明できる 関係、タプル、属性、ドメイン、演算子とそれを用いてリレーショナルデータモデルが説明できる リレーショナルモデルとデータベースとの関係が説明できる	リレーショナルモデル以外のデータモデルに関して説明できる
9	データベース言語に関して説明する  具体的例題を与えて、SQLの記述演習を行う	データ制御、データ定義、データ操作  インテグリティ制約、トリガー、プロシジャ SQLの標準化	SQLの概要を理解し、データ制御、データ定義、データ操作が行える  インテグリティ制約、トリガーとストアドプロシジャに関して説明できる SQLの標準化動向に関して理解している	
10	データベース設計の概要と論理設計を説明する  具体的例題を与えて、論理設計の演習を行う	データベース設計概要 論理設計とERモデル  正規化、ビュー設計 整合性制約	データベース設計の概要を説明できる 三層スキーマモデルを理解している ERモデルを利用し、論理設計を実行できる 正規化とビューを理解している	正規化が実行できる
11	物理設計に関して説明する	物理設計 トランザクション処理、排他処理 トランザクション分析 性能要件と、DBMSの機能、制約 インデックス、性能調整	物理設計の概要を理解している トランザクション処理、排他処理の概念を説明できる トランザクション分析を理解している データベース設計要件を説明できる。 インデックスに関して説明できる	トランザクション処理とDBの同期点の関係を説明できる トランザクション分析を利用できる  性能調整に関して留意するべき点を説明できる
12	データベースシステムの構成と障害回復に関して説明する	RDBMSの構成 障害回復機能	分散リレーショナルデータベースアーキテクチャを理解している ログ機能、バックアップリカバリ機能などRDBMSの障害回復機能に関して理解してい	クラスタ構成、レプリケーションに関して説明できる 階層ストレージ管理に関して説明できる
13	データベースの周辺技術に関して説明する	DWH、OLAP データマイニング XMLデータベース	データウェアハウス、オンライン分析処理を知っている データマイニングに関して基本的な手法を知っている XMLデータベースに関して知っている	データウェアハウス、オンライン分析処理を説明できる
14	重要事項のまとめ			
15	期末試験			
教科書・参考書	「オペレーティングシステムの基礎」 大久保英嗣 著 (サイエンス社) ISBN 4-7819-0860-8 「オペレーティングシステム」 清水謙多郎 著 (岩波書店) ISBN 4-00-007852-6 「オペレーティングシステム」 野口健一郎 著 (オーム社) ISBN 4-274-13520-1 「リレーショナルデータベース入門」 増永良文 著 サイエンス社 ISBN 4-781-91024-6 「データベースシステム」 北川博之 著 昭晃堂 ISBN 4-785-62046-3			
備考				

授業科目名		ネットワーク基礎			
単位数		2			
開設学期		2年後期			
目的		情報ネットワーク構成の基礎知識、具体的な事例に関する知識の獲得により、様々なアプリケーションにおいて、適切な情報通信技術を選択し利用することができる能力を身につけることを目的とする。			
概要		情報ネットワークを構成する様々な技術要素について基礎的な側面から具体的な応用例までを総合的に学ぶ。ネットワークを構成する各レイヤ技術についても学習する。			
目標					
先修科目					
関連科目					
授業方法		講義、机上演習、グループ演習			
評価方法・基準					
授業展開		学習成果	学習目標		
1	ネットワーク・インターネットの背景・歴史について説明する	CMP.cf.12	ネットワーク・インターネットの背景と歴史	ネットワーク・インターネットが発展してきた背景と歴史を踏まえ、その意義を説明することができる	
	普及しているネットワークアプリケーション事例について説明する		ネットワークアプリケーション	現在普及しているWebブラウザ、Eメール等様々なネットワークアプリケーションの理解を通して、社会におけるネットワークの重要性について説明することができる	
	ネットワークの基礎概念について説明する		ネットワークの基本概念	LAN/WAN/インターネット、トポロジー、通信プロトコル、スルーブット等、ネットワークにおける基本概念について説明することができる	
2	OSI参照モデルの概要とTCP/IPの位置づけについて説明する		OSI参照モデルとTCP/IP	プロトコルのレイヤー化の概念および重要性、またOSI参照モデルの概要とTCP/IPとの対応について説明することができる	
	OSI参照モデルの物理層について説明する		物理層	OSI参照モデルの物理層の位置づけ、また普及している伝送媒体の特徴やシャノンの法則等基盤となっている情報理論の概要、関連機器(リピータハブ)について説明することができる	物理層において基盤となっている物理理論の概要について理解する
	OSI参照モデルのデータリンク層について説明する		データリンク層	OSI参照モデルのデータリンク層の位置づけ、またEthernetにおけるCSMA/CD等、代表的なプロトコルの特徴、関連機器(ブリッジ、スイッチングハブ)について説明することができる	EthernetやPPP等のデータリンク層プロトコルの詳細について理解する
3	OSI参照モデルのネットワーク層とIPについて説明する		ネットワーク層およびIP	OSI参照モデルのネットワーク層の位置づけ、またIPにおけるパケット内の情報やIPアドレス、関連機器(ルータ)について説明することができる	IPv6において追加された概念等について理解する
	IPに関連したプロトコルについて説明する		IP関連プロトコル	NATの概念や、ICMP、ARP/RARP、DHCP等IPに関連したプロトコルについて説明することができる	
	ルーティングについて説明する		ルーティング	ルーティングの概念と、静的ルーティングと動的ルーティング、距離ベクトルアルゴリズムおよびリンク状態アルゴリズム等ルーティング方式の分類について説明することができる	OSPFやRIPなど代表的なルーティングプロトコルについて理解する
4	OSI参照モデルのトランスポート層とTCP/UDPの位置づけについて説明する		トランスポート層およびTCP/UDP	OSI参照モデルのトランスポート層の位置づけ、またポート番号の概念、TCPとUDPの違い、使い分けについて説明することができる	
	UDPについて説明する		UDP	UDPについて説明することができる	
	TCPについて説明する		TCP	TCPにおけるコネクション確立や開放、セッション、輻輳制御について説明することができる	
5	TCP/IPを用いたプログラミングについて説明する		TCP/IPプログラミング導入	TCP/UDPを用いたソケットプログラミングの考え方について説明することができる	
	TCP/IPを用いたプログラミング演習を行う		TCP/IPプログラミング演習	TCP/UDPを用いた単なるソケットプログラミングを行うことができる	TCP/UDPの差異について、実際のプログラムやその実行性能等の評価を通して、議論する
6	OSI参照モデルのセッション層、プレゼンテーション層、アプリケーション層について説明する		セッション層、プレゼンテーション層、アプリケーション層	OSI参照モデルのセッション層、プレゼンテーション層、アプリケーション層について説明することができる	
	様々なアプリケーションプロトコルの概要について説明する		アプリケーションプロトコル	クライアントサーバモデルおよび、HTTP、DNS、FTP、POP3、IMAP、SMTP等代表的なアプリケーションプロトコルの概要について説明することができる	
7	Webブラウザ・Webサーバの基本的な仕組みについて説明する		Webの仕組み	HTTPの詳細、Cookieやキャッシュの概念、URIやURLの概念等を用いて、Webアクセスの仕組みを説明することができる	
	Webにおいて利用されるプログラムの実行方式について説明する		Webにおけるプログラム実行方式	サーバーサイド、CGI、クライアントスクリプト、アプレット等のWebにおける様々なプログラムの実行方式について説明することができる	
8	HTTPを例としてクライアントサーバプログラミングについて説明する		クライアントサーバプログラミング導入	クライアントサーバプログラミングの考え方について説明することができる	
	HTTPを例としてクライアントサーバプログラミングの演習を行う		クライアントサーバプログラミング演習	クライアントサーバプログラミングを行うことができる	
9	HTTPの詳細や、FTPやPOP3等他のプロトコルを例として、クライアントサーバプログラミングについて説明する		クライアントサーバプログラミング導入	クライアントサーバプログラミングの考え方について説明することができる	
	HTTPの詳細や、FTPやPOP3等他のプロトコルを例として、クライアントサーバプログラミングの演習を行う		クライアントサーバプログラミング演習	クライアントサーバプログラミングを行うことができる	
10	セキュリティの基本概念について説明する		セキュリティの基本概念	セキュリティの基本概念(機密性、完全性、可用性)、および耐故障性やプライバシー等他の性質との位置づけの違いについて説明することができる	
	暗号・認証の仕組みについて説明する		暗号と認証	共有鍵暗号、公開鍵暗号のそれぞれの仕組みとこれらの使い分け、および電子署名や認証の仕組みについて説明することができる	代表的な暗号化や認証のアルゴリズムの概要について理解する
	暗号・認証の運用について説明する		暗号と認証の運用	PKI(鍵管理)、リプレイ攻撃、チャレンジアンドレスポンス、マンインザミドル攻撃等の概念と、暗号・認証に関する適切な運用方式の重要性を説明することができる	
11	システムセキュリティについて説明する		システムセキュリティ	認証(パスワード等)と認可(アクセス制御)、耐タンパ性、ウィルス、トロイの木馬等、個々のホスト上のOS、アプリケーションにおけるセキュリティの概念を理解し、様々な攻撃およびそれに対する技術上・運用上の対応策について説明することができる	
	ネットワークセキュリティについて説明する		ネットワークセキュリティ	IPSecやSSL、VPN等ネットワークプロトコルに関連し広く用いられている技術、およびクラッキング、ファイアウォール、IDS等のネットワークセキュリティに関する概念を理解し、様々な攻撃およびそれに対する技術上・運用上の対応策について説明することができる	
12	セキュリティ運用について説明する		セキュリティ運用	セキュリティポリシーの概念と重要性について説明することができ、システムのユーザとして、管理者として、または開発者として、それぞれどのような点に気をつけてセキュリティの実現に臨むべきなのか議論することができる	実際のシステム構成や運用ルール事例に対して、セキュリティ上の問題点、対応策について議論する
	セキュリティに関する事例について説明、議論する		セキュリティ事例	実際に起きた大きなインシデント、またバッファオーバーフローやクロスサイトスクリプティング、フィッシング等の典型的な問題について具体的に論じることにより、セキュリティの重要性や、技術上・運用上の課題について議論することができる	実際の動作コードに対して、セキュリティ上の問題点、対応策について議論する

授業展開	学習成果	トピック	学習目標	発展学習目標
13	関連技術や実用ツールについて説明する 最新のネットワーク技術について説明する	関連技術・実用ツール ネットワーク技術の最新動向	XMLや無線LAN等普及している関連技術, またWebサーバ, Webアプリケーションサーバ・フレームワーク等普及している関連ツールについて説明することができる ネットワーク技術の最新動向について説明することができる	普及しているツール等を用いたネットワーク構築・管理やアプリケーション構築の演習を行う 最新のツール等を用いたネットワーク構築・管理やアプリケーション構築の演習を行う
14	重要事項のまとめ			
15	期末試験			
教科書・参考書	「マスタリングTCP/IP 入門編」(竹下隆史・村山公保・荒井 透・苅田幸雄, オーム社), 「マスタリングTCP/IP 応用編」(Philip Miller著, 苅田 幸雄監訳, オーム社), 「コンピュータネットワーク」(Andrew S. Tanenbaum 著, 日経BP社), 「情報セキュリティ技術大全:信頼できる分散システム構築のために」(Ross Anderson著, トップスタジオ訳, 日経BP社)			
備考				

授業科目名	工学基礎				
単位数	2				
開設学期	4年生前期				
目的	社会に出て活躍するために専門性の高い技術に加え、一般的な工学の概念を身に付け、物事を論理的に深く洞察し問題に対しては解決手法を提示できるようにすることを目的とする				
概要	問題解決技法、統計解析、測定とマトリクス、システム特性、工学的設計の基本概念、測定の理論、倫理綱領とプロフェッショナルとしての行動姿勢を学ぶほか、工学原理についても触れる				
目標	問題解決の技法を知り、事象をシステムティックに捉え、問題を定式化し、解決の手法を見出し、分析、評価して解決することができる				
先修科目					
関連科目					
授業方法	講義				
評価方法・基準	下記により評価する。 ・期末試験の結果				
授業展開	学習成果	トピック	学習目標		
1	授業の概要と目的、進め方、評価基準について説明する(0.3h)	FND.ef.5	工学の定義と意義	FND.ef.5	自然科学の成果を適用して、生産(「ものづくり」でもよい)を品質・効率良く行うという一般的な工学の定義が理解できる
	工学的設計の基本概念について解説する(1)(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.5	工学の歴史	FND.ef.5	古代ギリシャ・ローマの(軍事及び)土木から始まり、ソフトウェア、金融、教育のような無形物やサービスを扱うまでになるに至った工学の歴史について触れる
		FND.ef.7	工学の主要分野	FND.ef.7	機械工学、電気工学、土木工学、航空工学、化学工学などの工学の主要分野とその扱う内容について触れる
		FND.ef.5	工学と社会	FND.ef.5	工学の社会的使命、および職業倫理の必要性が理解できる
2	工学的設計の基本概念について解説する(2)(講義冒頭0.1hで本日の講義の概要を説明)	MAA.af.1	まとまりの分析(完全性、一貫性、ロバスト性など)	MAA.af.1	まとまりの分析(完全性、一貫性、ロバスト性など)を理解し使用できる
		MAA.af.2	正当性の分析(静的解析、シミュレーション、モデルチェックなど)	MAA.af.2	正当性の分析(静的解析、シミュレーション、モデルチェックなど)が理解できる
		MAA.af.3	品質特性(非機能特性)の分析(安全性、セキュリティ、性能、根本原因分析など)	MAA.af.3	品質特性(非機能特性)の分析(安全性、セキュリティ、性能、根本原因分析など)を理解し使用できる
		MAA.af.4	優先順位付け、トレードオフの分析、要求のためのリスク分析及び影響解析	MAA.af.4	優先順位付け、トレードオフ分析、要求のためのリスク分析および影響解析が理解できる
		MAA.af.5	トレーザビリティ	MAA.af.5	トレーザビリティを理解し使用できる
3	工学的設計の基本概念について解説する(3)(講義冒頭0.1hで本日の講義の概要を説明)	FND.ef.5	工学的設計の定義	FND.ef.5	工学的設計は、自然科学で得られた知見を満たすことが理解できる
		FND.ef.5	工学的設計の実践	FND.ef.5	工学的設計は、工学実践の経験に基づくことが理解できる
		FND.ef.5	工学的設計の制約	FND.ef.5	工学的設計は、品質、時間、費用、空間、環境などの制約を充足することが理解できる
4	問題解決技法とは何か、代表的な問題解決サイクルについて解説する(講義冒頭0.1hで本日の講義の概要を説明)	CMP.cf.3	問題解決の必要性	CMP.cf.3	問題を定義し解決方法を探り、解決方法を比較評価するのが工学的問題解決方法であることが理解できる
		CMP.cf.3	問題解決技法と本講義の関係	CMP.cf.3	本講義と問題解決技法の関係が理解できる(マトリクスなどの比較評価技法を本講義では扱う)
		CMP.cf.3	問題解決のプロセス	CMP.cf.3	工学的問題解決のプロセス(問題の構造化、見える化)が理解できる
		CMP.cf.3	問題解決のための準備活動	CMP.cf.3	問題解決のための準備活動が理解できる
		CMP.cf.3	問題解決のための環境構築	CMP.cf.3	問題解決を成功に導くチームビルディングやコミュニケーションなどについて理解できる
		CMP.cf.3	問題解決事例	CMP.cf.3	実際の問題解決事例を聞いて、問題解決のサイクルがイメージできる
5	ソフトウェア開発と問題解決技法との関係を説明する	CMP.cf.3	問題事例	CMP.cf.3	ソフトウェア開発における問題事例が理解できる
		CMP.cf.3	問題解決事例	CMP.cf.3	各事例に対して、解決方法をグループで話し合う
		CMP.cf.3	ソフトウェア開発における問題解決事例	CMP.cf.3	プログラミング言語による問題解決、マネジメントによる問題解決など、各事例に対する典型的な解決技法を紹介する
6	システムの基礎(システムとは何か、システムにはどのようなものがあるか)について解説する(講義冒頭0.1hで本日の講義の概要を説明)	CMP.cf.6	システム概念	CMP.cf.6	システムとは何かが理解できる
		CMP.cf.6	システムの分類	CMP.cf.6	システムの分類について理解できる

授業展開	学習成果	トピック	学習目標
	CMP.cf.6	情報システムの構成	CMP.cf.6 システムとハードウェア、ソフトウェア、ネットワークの関係が理解できる
	CMP.cf.6	情報システム事例	CMP.cf.6 組み込みソフトウェア、エンタプライズソフトウェアの身近な例が把握できる
7	FND.ef.1	統計的技法と伝統的問題解決技法	FND.ef.1 Six Sigmaなどの統計的技法と伝統的問題解決技法の相違が理解できる。
	FND.ef.1	統計的技法のプロセス	FND.ef.1 統計的技法のプロセスが理解できる。
8	FND.ef.6	実世界の現象	FND.ef.1 測定誤差やノイズを含む実世界の現象に対する確率的なモデル化が理解できる
	FND.ef.6	現実と統計分析結果のギャップ	FND.ef.1 市販のソフトウェアによるブラックボックス的な統計処理による表面的な成果と現実とのギャップが理解できる
	FND.ef.6	確率統計の概念	FND.ef.1 数値データから算出されるもの(確率、平均、分散)と、確率変数としてモデル化し分布の概念を通して定めたものとの関連や相違点が理解できる
	FND.ef.6	統計的分析手法	FND.ef.1 基本的な統計データを解析する場合に必要な概念が理解できる(例:正規分布から派生する。2乗分布、t分布、F分布などが、どのような場面で用いられるかを理解して、それらに基づく推定や検定の方法が理解できる)
9	FND.ef.6	測定の定義と意義	FND.ef.6 測定とは何か、測定の意義が理解できる
	FND.ef.6	測定誤差と有効数字	FND.ef.6 測定誤差、有効数字、デシベル表示について理解できる
10	FND.ef.3	ソフトウェア測定	FND.ef.3 ソフトウェア開発において、品質を作り込むための手法を知り、ソフトウェアの「品質」を定義する意義が理解できる
	FND.ef.3	プロダクトメトリクス	FND.ef.3 プロダクトメトリクスについて理解できる
	FND.ef.3	プロダクト品質メトリクス	FND.ef.3 プロダクト品質メトリクスについて理解できる
	FND.ef.3	プロセスメトリクス	FND.ef.3 プロセスメトリクスについて理解できる
11	FND.ef.4	システム構成の決定要因	FND.ef.4 情報システムを構成する要素について理解できる
	FDN.ef.4	ISOが定めるソフトウェア製品の内部品質及び外部品質の特性	FND.ef.4 ISO/IEC9126で定められているソフトウェア製品の品質(機能性、信頼性、使用性、効率性、保守性、移植性)について理解できる。利用時の品質として、有効性、生産性、安全性、満足性が定められていることにも触れる
12	FND.ef.7	工学的問題解決とモデル化	FND.ef.7 工学的問題解決において、モデリングに自然科学の知見が活用されることが理解できる
	FND.ef.7	工学的問題解決と自然科学	FND.ef.7 工学的問題解決において、解の検証・選択に自然科学の知見が活用されることが理解できる
13	PRF.pr.2	倫理の定義	PRF.pr.2 技術者が担う責任、技術者に求められる倫理の変化について理解できる
	PRF.pr.2	法令順守と社会的責任	PRF.pr.2 意思決定の際に、様々な利害関係者、利用者、未来の影響者までを視野に入れた多面的な価値を認識する必要性を理解できる
	PRF.pr.2	設計段階で守るべき倫理	PRF.pr.2 技術者としての特性を活かしながら、倫理の問題を分析する手法を理解できる
	PRF.pr.2	法令に沿った設計の進め方	PRF.pr.2 業務遂行の局面で遭遇する倫理問題の事例演習を通して、経営に資する実践的な考え方が理解できる
14		重要事項のまとめ	
15		期末試験	
教科書・参考書	・仙場洋一 著「信頼を得る 上級SEの問題解決力-SEが直面する33の問題シーン」 (技術評論社、ISBN 4-7741-1817-6) ・宮川宏之 著「どこでどう使うQC手法」 (プレーン・ダイナミクス、978-4-938340-28-5 (4-938340-28-3)) ・ローレンス・H.パトナム/著 ウェア・マイヤーズ/著 山浦恒央/訳「初めて学ぶソフトウェアメトリクス」 (日経BP社、978-4-8222-8242-4 (4-8222-8242-2)) ・ロジャー・プレスマン: 実践ソフトウェアエンジニアリング, 日科技連, 2005. ・エリック・プロディ: ソフトウェアエンジニアリング, 翔泳社, 2004 ・Frye, Ellen., Engineering Problem Solving for Mathematics, Science, and Technology Education (http://engineering.dartmouth.edu/teps/book.html)に目次あり。 ・Dym, C. L., and Little, M. (2000): Engineering Design: A Project-Based Introduction, John Wiley & Sons, New York		
備考			

授業科目名	ソフトウェア構築
単位数	2
開設学期	
目的	ソフトウェアを構築するにあたり必要とされる過程を理解し、構築の際に用いられる技術について学習することを目的とする。
概要	ソフトウェアの構築とは何を行うことかを習得する。ソフトウェアを構築するに当たり、必要とされる過程とその重要性、および構築のための要素技術を習得する。
目標	作成されるソフトウェアの設計内容が与えられた際、自分で、あるいは多数で共同してソフトウェアを構築する能力を身につける。ソフトウェアを構築する他の作業者に、ソフトウェアを構築するために必要な作業が何かを指示し管理する能力を身につける。
先修科目	
関連科目	ソフトウェア設計とHCI、開発プロセスと保守、ソフトウェア開発マネジメント、ソフトウェアアーキテクチャ
授業方法	講義
評価方法	期末試験の成績により決定する。
基準	
授業展開	

1	授業の概要と目的および進め方について説明する。	学習成果	トピック	学習目標	発展学習目標
	ソフトウェア開発の全体的な流れについて説明する。また、その流れの中でソフトウェア構築がどのような位置づけであるか説明する。	PRO.co n	プロセスの基礎	PRO.con	ソフトウェア開発プロセスとは何かを説明できる。ソフトウェア開発プロセス中において、ソフトウェア構築がどのような位置づけであるかを説明できる。
	ソフトウェア開発の要素としてどのような作業があるかを理解する。	PRO.co n	プロセスの記述	PRO.con	ソフトウェア開発プロセスにはどのような作業が必要か、自分で考えて列挙することができる。
2	ソフトウェア開発の全体的な流れの中で、ソフトウェア構築がどのような位置づけであるか、ソフトウェア設計との関連から説明する。	DES.con	設計に用いられる概念	DES.con	ソフトウェアの設計とは何かを説明できる。ソフトウェアの設計段階と構築段階の違い、関連性を説明できる。
3	ソフトウェア構築における開発環境の役割について説明する。	CMP.tl.1	開発環境	CMP.tl.1	開発環境とは何かを説明できる。開発環境により、ソフトウェア構築がどのように行われるかを理解する。
		CMP.tl.1	開発環境の認識	CMP.tl.1	開発環境で使われる各種ツールがどのように使われているか、いままで使った経験のあるツールはどのようなものであるかを列挙することができる。
4	ソフトウェア設計時に用いるプログラミング言語としてオブジェクト指向言語を取り上げ、その特徴について説明する。	CMP.ct.3	オブジェクト指向パラダイムにおける実行時のトピック(ポリモルフィズム、ダイナミックバインディングなど)	CMP.ct.3	ポリモルフィズム、ダイナミックバインディングが何かを説明できる。あるオブジェクト指向プログラミング言語を対象とし、ポリモルフィズムやダイナミックバインディングなどがどのような目的で用いられているかを理解し、ソフトウェア構築の際に利用することができる。
		CMP.ct.4	パラメータ化と汎化	CMP.ct.4	パラメータ化、汎化とは何かを説明できる。あるオブジェクト指向プログラミング言語を対象とし、パラメータ化や汎化がどのような目的で用いられているかを理解し、ソフトウェア構築の際に利用することができる。
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング	CMP.ct.5	アサーション、契約による設計、防御的プログラミングが何かを説明できる。あるオブジェクト指向プログラミング言語を対象とし、アサーション、契約による設計、防御的プログラミングがどのような目的で用いられているかを理解し、ソフトウェア構築の際に利用することができる。
		CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラント	CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラントが何かを説明できる。あるオブジェクト指向プログラミング言語を対象とし、エラーハンドリング、例外処理をソフトウェア構築の際に利用することができる。フォールトトレラントを実現するためにどのようなソフトウェア構築技術が用いられているかを理解する。
5	GUI構築ツールを用いることにより、ソフトウェア構築をどのように行うことができるかを説明する。	CMP.tl.2	GUI構築ツール	CMP.tl.2	GUI構築ツールが何かを説明できる。GUI構築ツールを用いることにより、ソフトウェア構築のうちどのような作業をどのように手助けすることが可能であるか理解する。
	単体テストツールを用いることにより、ソフトウェア構築をどのように行うことができるかを説明する。	CMP.tl.3	単体テストツール	CMP.tl.3	単体テストツールが何かを説明できる。単体テストツールを用いることにより、ソフトウェア構築のうちどのような作業をどのように手助けすることが可能であるか理解する。
		CMP.tl.3	単体テストツールの利用	CMP.tl.3	単体テストツールを利用し、テストを実行することができる。
		CMP.ct.16	テストファーストプログラミング	CMP.ct.16	テストファーストプログラミングとはなにか説明できる。あるオブジェクト指向プログラミング言語を対象とし、テストファーストプログラミングの考え方をソフトウェア構築の際に利用することができる。
6	複数の実行が並列・並行に動作するソフトウェアの動作原理について説明する。また、そのようなソフトウェアを構築するための技術について説明する。	CMP.ct.10	並列処理の基本要素(セマフォ、モニターなど)	CMP.ct.10	セマフォ、モニターが何かを説明できる。
		CMP.ct.12	分散ソフトウェアのための構築技術	CMP.ct.12	分散ソフトウェアとは何かを説明できる。分散ソフトウェアが用いられているアプリケーション分野と、その分野における利用例を理解する。
7	ソフトウェア設計の際に状態やテーブルが用いられた際、それをソフトウェア構築時どのようにして実装することができるかを説明する。	CMP.ct.7	状態ベースおよびテーブル駆動の構築技法	CMP.ct.7	状態ベースのソフトウェア構築技法、および、テーブル駆動のソフトウェア構築技法とはなにかを説明できる。状態ベースやテーブル駆動の技法を用いたソフトウェアを読み、どのように用いられているかを理解できる。
		CMP.ct.7	状態ベースおよびテーブル駆動を用いたプログラムの作成	CMP.ct.7	状態ベースあるいはテーブル駆動のソフトウェア構築技法を用いてプログラムを作成することができる。
8	ソフトウェアの振舞いをその実行時に決定する方法について説明する。また、その一例としてソフトウェアの国際化を取り上げ、国際化がどのようにして行われるかを説明する。	CMP.ct.8	実行時コンフィグレーションと国際化	CMP.ct.8	実行時コンフィグレーションとは何か説明できる。ソフトウェアの国際化とは何かを説明できる。
9	文字列等を入力とするソフトウェアを対象とし、入力内容をソフトウェア内部で利用するために必要となる処理について説明する。	CMP.ct.9	文法ベースの入力処理(パース処理)	CMP.ct.9	文法ベースの入力処理方法とは何か説明できる。文法ベースの入力処理を行うソフトウェアを読み、どのように用いられているかを理解できる。
		CMP.ct.9	文法ベースの入力処理を行うプログラムの作成	CMP.ct.9	文法ベースの入力処理を行う簡単なプログラムを作成することができる。
10	実世界のソフトウェアにおける特定のドメインに関するソフトウェアについて説明する。	CMP.ct.11	ミドルウェア(コンポーネントとコンテナ)	CMP.ct.11	ミドルウェアとは何かを説明できる。また、ミドルウェアにおいて、コンポーネントとコンテナとは何か、その関係について説明できる。
		CMP.ct.13	組込みシステムの構築とハードウェア・ソフトウェア協調設計	CMP.ct.13	組込みソフトウェアとは何か説明できる。ハードウェア・ソフトウェア協調設計とは何か説明できる。組込みソフトウェアの構築における問題を理解する。ハードウェア・ソフトウェア協調設計の必要性を理解する。
				CMP.ct.11	ミドルウェアを用いる簡単なソースコードを作成することができる。
				CMP.ct.13	簡単な組込みソフトウェアを作成することができる。

授業展開	学習成果	トピック	学習目標	発展学習目標
11	ソフトウェアを構築する段階における、ソフトウェア分析手法について説明する。	CMP.ct.14 ホットスポット分析とパフォーマンスチューニング	CMP.ct.14 ホットスポット分析とは何かを説明できる。あるオブジェクト指向プログラム言語で書かれたソフトウェアを対象とし、パフォーマンスチューニングを行うことによって、ソフトウェアの実行速度がどの程度変化するかを理解する。	CMP.ct.14 与えられたソースコードに対してホットスポット分析を行うことにより、パフォーマンスチューニングを実際に行うことができる。
		CMP.tl.5 プロファイリング・パフォーマンス分析・スライシングのツール	CMP.tl.5 プロファイリング、パフォーマンス、スライシングとは何かを説明できる。プロファイリングやパフォーマンス分析によって得られる、ソフトウェア構築の際の利点について理解する。スライシングを行うためのツールとその効果を理解する。	CMP.tl.5 与えられたソースコードに対してプロファイリングやパフォーマンス分析を実際に行うことができる。スライシングツールを用いてスライシングを実行することができる。
		CMP.ct.14, CMP.tl.5 静的解析と動的解析	CMP.ct.14, CMP.tl.5 ソースコードに対する静的解析、動的解析とは何かを説明できる。あるオブジェクト指向プログラミング言語で書かれたソフトウェアを対象とし、静的解析や動的解析を行うことによって得られる情報が何かを理解する。静的解析と動的解析の違いや特徴を説明できる。	CMP.ct.14, CMP.tl.5 与えられたソースコードに対して静的解析や動的解析を実際に行うことができる。
12	ソフトウェアを構築する際、既存のソフトウェアを流用することによって効率的にソフトウェアを作成する方法について説明する。また、既存のソフトウェアを流用することによる効果について説明する。	CMP.ct.15 コーディング規約	CMP.ct.15 コーディング規約とは何かを説明できる。コーディング規約を用いることによる利点について説明できる。	CMP.ct.15 与えられたコーディング規約に従うよう、自分の書いたソースコードを修正することができる。
		CMP.ct.1 APIの設計と利用	CMP.ct.1 APIとはなにか説明できる。あるオブジェクト指向プログラミング言語で書かれたソフトウェアを対象とし、そのソフトウェアでどのようなAPIが定義されているかを把握できる。また、そのAPIを用いてソフトウェアを構築することができる。	CMP.ct.1 APIを用いたプログラムを作成することができる。
		CMP.ct.2 コードの再利用とライブラリ	CMP.ct.2 コードの再利用、ライブラリとは何か説明する。あるオブジェクト指向プログラミング言語を対象とし、既存のソースコードを再利用、あるいは、既存のライブラリを用いたソフトウェアを構築することができる。	CMP.ct.2 自分が作成したプログラムの一部を、既存の与えられたソースコード等を利用することによって再構築することができる。
		CMP.ct.15 プラットフォーム標準(POSIXなど)	CMP.ct.15 POSIX標準とはなにか説明できる。POSIX標準にのっとったソフトウェア構築を行うことの重要性について理解する。	CMP.ct.15 POSIX標準にのっとっていないソースコードを、POSIX標準にのっとった形で書き換えることができる。
13	ソフトウェア構築をより抽象度の高いレベルで構築するとき用いられるプログラミング言語について説明する。	CMP.tl.4 アプリケーション指向言語(スクリプト言語、ビジュアル言語、ドメイン特化言語、マークアップ言語、マクロなど)	CMP.tl.4 アプリケーション指向言語とは何か説明できる。アプリケーション指向言語が得意とするようなソフトウェア分野について理解する。	CMP.tl.4 アプリケーション指向言語を用いて、簡単なソースコードを作成することができる。
14	重要事項のまとめ			
15	期末試験			
教科書・参考書				
備考				



授業科目名	<b>モデル化と要求開発</b>
単位数	2
開設学期	2年生後期
目的	ソフトウェアのモデル化と分析のための概念を理解し、良いモデルを構成する能力と良い要求仕様を構成する能力を獲得することを目的とする
概要	ソフトウェア開発においては、開発対象のモデル化と要求仕様の作成が必須であり、これらのプロダクトの質がソフトウェア開発の成否に関わってくる。本講義では、モデル化の原則、モデルの種類、モデルの分析基礎、要求分析の基礎、要求の獲得、要求の仕様化、要求の妥当性確認といった一連のモデル化技法と要求定義技法について学ぶ。
目標	ソフトウェア開発者として必要な能力であるソフトウェアモデル化技法と要求定義技法を習得する。具体的には各種モデルを種々の観点から分析できる能力、要求を獲得し高品質な要求を仕様化する能力、要求の妥当性を確認する能力の習得を目標とする
先修科目	
関連科目	
授業方法	
評価方法・基準	
授業展開	

1	学習成果	トピック	学習目標	発展学習目標	
1	授業の概要と目的および進め方について説明する(※講義0.3h)。モデルの概念と意義、モデル化の原則について説明する。	MAA.md.1, MAA.md.5	モデル化という概念の定義と意義	MAA.md.1, MAA.md.5	モデル化の概念を説明できる
		MAA.md.1	基本的なモデル化の原則(抽象化、汎化、集約、投影、ビューなど)	MAA.md.1	モデル化の原則を説明できる
2	要求工学について概説し、要求分析のプロセス、要求のレベルについて説明する	MAA.rfd.1	ソフトウェア工学と要求工学の関係、要求工学の位置づけ、要求の定義	MAA.rfd.1	要求工学の位置づけを説明できる
		MAA.rfd.2	要求獲得、仕様化、仕様の精製、仕様の管理	MAA.rfd.2	要求定義プロセスを説明できる
		MAA.rfd.3	ニーズ、ゴール、ユーザ要求、システム要求、ソフトウェア要求など	MAA.rfd.3	要求の種類を説明できる
3	要求仕様の満たすべき品質特性とその分析について述べる	MAA.rfd.4	妥当性、非あいまい性、無矛盾性、完全性、検証可能性、変更可能性、優先度、追跡可能性に関する説明	MAA.rfd.4	要求仕様の品質特性を説明できる。一定水準の品質特性を満たす仕様が記述できる
		MAA.rfd.5	安全性、セキュリティ、使用性、性能	MAA.rfd.5	最終ソフトウェア製品が満たすべき品質特性を説明できる
		MAA.rfd.6, MAA.rfd.7	優先度、トレードオフ、リスク、要求とアーキテクチャの競合に関する分析	MAA.rfd.6, MAA.rfd.7	優先度、トレードオフ、リスク、要求とアーキテクチャの競合について具体例を挙げて説明できる
4	要求仕様の評価法と構成法について述べる	MAA.rv.4	要求仕様の8つの品質特性の評価	MAA.rv.4	要求仕様の品質特性を評価できる。
		MAA.rsd.1	要求仕様の構成、読み手、種類	MAA.rsd.1	要求仕様の構成や種類を説明できる。読み手を考慮する必要性を説明できる
		MAA.rsd.2	IEEE std 830に基づいた要求仕様の構成法	MAA.rsd.2	要求仕様を構成できる
5	レビューとインスペクションについて述べる。また要求仕様の妥当性を確認する手法としてプロトタイプングについて述べる	MAA.rv.1	レビューとインスペクションの概要、効果	MAA.rv.1	レビューとインスペクションの差異を説明できる。また効用を説明できる
		MAA.rv.2	ラビッドプロトタイプングの概要と効果	MAA.rv.2	プロトタイプングの効用を説明できる
		MAA.rv.3	要求仕様からのテスト項目の設計	MAA.rv.3	受け入れテストについて説明できる
6	要求獲得の対象と獲得技法の中で問題分析手法とインタラクション分析法、合意形成法について述べる	MAA.er.1	ステイクホルダ、ドメインエキスパート、操作環境、組織環境など	MAA.er.1	要求を獲得する対象について説明できる
		MAA.er.2	問題分析、発想法、整理法	MAA.er.2	主な問題分析法の手順を説明できる
		MAA.er.2	インタビュー、会議、インタラクション分析	MAA.er.2	インタラクションとその分析法について説明できる
		MAA.tm.8	ビューポイント、要求の競合・矛盾、AHP法	MAA.tm.8	ビューポイントと要求の競合・矛盾の生じる原因、解消法について説明できる
7	ゴール指向分析と要求獲得プロセス全般を扱う技法を紹介する	MAA.er.3	ゴール、ゴール指向分析	MAA.er.3	ゴール指向分析の手順を説明できる
		MAA.er.3	ソフトウェア要求獲得技法	MAA.er.3	要求獲得技法の手順を説明できる
8	要求をモデル化・仕様化するための言語(要求言語)について述べる	MAA.rsd.3	構造化言語、図式言語(IDEF0,UML)	MAA.rsd.3	構造化言語や図式言語によるモデルを理解できる
		MAA.rsd.3	Z, OBJ, VDM	MAA.rsd.3	形式言語による簡単なモデルを理解できる
9	情報やデータ、システムの静的な構造のモデル化について述べる(1)	MAA.tm.1	実体関連図	MAA.tm.1	実体関連図によるモデルを理解できる
		MAA.tm.1	クラス図他	MAA.tm.1	クラス図によるモデルを理解できる
10	情報やデータ、システムの静的な構造のモデル化について述べる(2)	MAA.tm.3	ソフトウェアアーキテクチャ	MAA.tm.3	ソフトウェアアーキテクチャの概念を説明でき、モデルを理解できる
		MAA.tm.5	コンポーネント図	MAA.tm.5	コンポーネント図によるモデルを理解できる

授業展開	学習成果	トピック	学習目標	発展学習目標
11 対象の動的な振舞いのモデル化について述べる(1)	MAA.tm.2	DFD,DD,プロセス仕様	MAA.tm.2 構造化分析の手順を説明でき、モデルを理解できる	MAA.tm.2 構造化分析を実践して、DFDやDD,プロセス仕様書のモデル化ができる
	MAA.tm.2	シーケンス図	MAA.tm.2 シーケンス図によるモデルを理解できる	MAA.tm.2 シーケンス図によるモデル化ができる
	MAA.tm.2	状態遷移図	MAA.tm.2 状態遷移図によるモデルを理解できる	MAA.tm.2 状態遷移図によるモデル化ができる
12 対象の動的な振舞いのモデル化について述べる(2)	MAA.tm.2	ユースケース図、ユースケース記述、シナリオ	MAA.tm.2 ユースケース図、ユースケース記述、シナリオによるモデルを理解できる	MAA.tm.2 ユースケース図、ユースケース記述、シナリオによるモデル化ができる
	MAA.rm.2	FMEA, FTA	MAA.rm.2 FMEA, FTAによるモデルを理解できる	MAA.rm.2 FMEA, FTAによるモデル化ができる
13 ドメイン、エンタープライズシステム、組み込みシステムのモデル化手法について述べる	MAA.tm.4	ドメイン工学アプローチ	MAA.tm.4 ドメイン工学について説明できる	MAA.tm.4 ドメイン分析を実践してドメインのモデル化ができる
	MAA.tm.6	ビジネスプロセス、ビジネスゴール、組織	MAA.tm.6 ビジネスプロセスやビジネスモデルについて説明できる	MAA.tm.6 エンタープライズシステムのモデル化ができる
	MAA.tm.7	リアルタイムスケジュール分析、外部インタフェース分析	MAA.tm.7 リアルタイムスケジュール分析や外部インタフェース分析によるモデルを理解できる	MAA.tm.7 組み込みシステムのモデル化ができる
14 アナリシスパターン、まとめ	MAA.tm.9	アナリシスパターン	MAA.tm.9 アナリシスパターンについて説明できる	MAA.tm.9 アナリシスパターンをモデル化できる
	MAA.tm.9	プロブレムフレーム	MAA.tm.9 プロブレムフレームについて説明できる	MAA.tm.9 プロブレムフレームをモデル化できる
15 期末試験				
教科書・参考書	要求工学(大西・郷著)、共立出版			
備考				

授業科目名	ソフトウェアアーキテクチャ
単位数	2
開設学期	3年生前期
目的	設計の概念、設計にて考慮すべき事柄(品質特性、技法、ライフサイクル等)と代表的なアーキテクチャのパターンについて学習することを目的とする。
概要	ソフトウェアの設計とは何かを習得する。ソフトウェア設計において、考慮すべき品質特性、設計技術、戦略、設計の表現形態、支援ツール、設計の評価方法について習得する。
目標	設計に用いられる概念、設計のパラダイム、アーキテクチャ設計のスタイル、設計の支援ツールと評価方法について、基本概念を習得し、実際の設計課題に直面した際に、適切な解決策を導き出すための基礎能力を身につける。
先修科目	
関連科目	
授業方法	講義、机上演習、グループ演習 ・設計手法、表記については机上演習を行う。 ・設計特性やアーキテクチャで考慮すべき事柄については、グループ演習でディスカッションさせる。そのために、あらかじめ3~4名のグループを設定しておく。設計事例(身近なシステム:WebショッピングやWebバンキング)を用意し、「グループ演習」の際に、割り当てられた課題に対して、講義で説明された各種概念が、具体的にどのような事柄にあたるのかをグループでディスカッションさせ、発表させる。
評価方法・基準	下記により評価する。 ・机上演習による提出物 ・グループ演習での発表結果 ・中間および期末試験の結果
授業展開	

1	学習成果	トピック	学習目標	発展学習目標
1	授業の概要と目的および進め方について説明する。必要に応じ、グループ演習用のグループ分けを発表する。(※講義0.3h)。	DES.con.1 設計という概念の定義	DES.con.1 設計という概念(一般的な定義、開発プロセス内での位置づけ、役割、主要アウトプット)を説明できる。	DES.con.1 具体的例によるソフトウェアの設計ができる。具体例に基づいて、ソフトウェアの設計とは何かを説明できる。
	設計の概念、基本的な考慮事項、開発ライフサイクルにおける位置づけ、設計の原則、要求・品質との関係を説明する。	DES.con.2 基本的な設計の考慮事項(データの永続性、ストレージマネジメント、例外など)	DES.con.2 ソフトウェア設計における主要な問題(並行処理、イベント制御と処理、コンポーネントの分散化、エラー及び例外、対話と表示、データの永続化等)の概念を説明できる。	DES.con.2 設計の考慮事項を満足するようにソフトウェアを設計できる。ソフトウェアが設計の考慮事項を満足している、あるいは、していないことを評価できる。
		DES.con.3 複数のソフトウェア開発ライフサイクルにおける設計の関係	DES.con.3 ソフトウェア開発ライフサイクルにおいて、設計の持つ役割を説明できる。要求定義と設計の関係、設計とテストとの関係を説明できる。	DES.con.3 要求定義結果に基づいて、ソフトウェアを設計できる。また設計した結果が、詳細設計・テスト可能な成果物となるように設計できる。
		DES.con.4 設計の原則(情報隠蔽、凝集度と結合度)	DES.con.4 設計の原則(情報隠蔽、凝集度と結合度)の概念を説明できる。設計の原則が満足あるいは満足されない場合にソフトウェア構造にどのような影響を与えるかを説明できる。	DES.con.4 設計の原則を満足するようにソフトウェアを設計できる。ソフトウェアが設計の原則を満足している、あるいは、していないことを評価できる。
		DES.con.5 設計と要求との競合	DES.con.5 要求定義に対する設計との関係、設計結果に対して要求を満足していることの妥当性の検証の必要性などの概念を説明できる。	DES.con.5 設計結果に対して、要求を満足していることの妥当性の検証ができる。
		DES.con.6 品質特性の設計(信頼性、ユーザビリティ、性能、テスト容易性、フォールトトレラント性)	DES.con.6 品質の良いソフトウェアを設計するために考慮すべき事柄には何があるか、またその各事柄の概念の定義、設計においてどのような影響があるのかが説明できる。	DES.con.6 品質特性を考慮したソフトウェアの設計ができる。ソフトウェアが品質特性を満足している、あるいは、していないことを評価できる。
2	設計におけるトレードオフ、アーキテクチャのスタイル、パターンなどの考え方があることを説明する。	DES.con.7 設計におけるトレードオフ	DES.con.7 設計には様々な戦略(分割・統治、段階的詳細化、データ抽象化・情報隠蔽、反復、インクリメンタル等)と手法(機能指向、オブジェクト指向、データ中心、コンポーネント指向、アスペクト指向等)があり、それぞれに対する特性、適用条件、利点・欠点などの基本概念を説明できる。	DES.con.7 設計における様々な戦略を考慮して、ソフトウェアを設計できる。選択した戦略の妥当性を説明できる。
	ソフトウェア設計で説明した設計手法の振り返り(その1)設計手法の全体像、機能指向、オブジェクト指向、データ中心、アスペクト指向による設計手法の関係について説明する。さらに、機能指向による設計手法について説明する。	DES.con.8 アーキテクチャのスタイル、パターン、再利用	DES.con.8 アーキテクチャの基本概念を説明できる。アーキテクチャを表現するための視点、スタイル、パターン、再利用におけるアーキテクチャの役割を説明できる。	DES.con.8 アーキテクチャの基本概念に基づいてソフトウェアを設計できる。適用したスタイル、パターンの妥当性を説明できる。
		DES.str.1 設計手法の全体像	DES.str.1 ソフトウェア設計手法(定義、考え方、プロセス、主要成果物)にはどのようなものがあるか説明することができる。	DES.str.1 機能指向、オブジェクト指向、データ中心設計、アスペクト指向などの手法を適材適所で組み合わせ、要求定義から導出された仕様に基づきソフトウェアの設計ができる。設計結果に対して、適用した手法の妥当性を評価できる。
		DES.str.1 機能指向による設計	DES.str.1 機能指向による設計手法の概念(定義、考え方、プロセス、主要成果物)について説明できる。基本的な要求に対して、定義されたプロセスに従って設計成果物を作成することができる。	DES.str.1
3	ソフトウェア設計で説明した設計手法の振り返り(その2)オブジェクト指向による設計手法、設計結果の記法について説明する。オブジェクト指向による設計の発展形としてアスペクト指向による設計手法について説明する。	DES.str.2 オブジェクト指向による設計	DES.str.2 オブジェクト指向による設計手法の概念(定義、適用条件、考え方、プロセス、主要成果物)について説明できる。基本的な要求に対して、定義されたプロセスに従って設計成果物を作成することができる。	DES.str.2 機能指向、オブジェクト指向、データ中心設計、アスペクト指向などの手法を適材適所で組み合わせ、要求定義から導出された仕様に基づきソフトウェアの設計ができる。設計結果に対して、適用した手法の妥当性を評価できる。
		DES.str.4 アスペクト指向による設計	DES.str.4 アスペクト指向による設計手法の概念(定義、適用条件、考え方、プロセス、主要成果物)について説明できる。	DES.str.4
		DES.ar.6 アーキテクチャのための記法(アーキテクチャ上のビューポイントと表現、コンポーネント図など)	DES.ar.6 ソフトウェア設計成果物を表現するための代表的な記法(名称、役割、読み方、書き方)が説明できる。構造と振舞いに関して、1つ以上の記法によって、設計結果を記述できる。	DES.ar.6 代表的な記法を利用したソフトウェアの設計ができる。複数の記法によって、同一の要求の設計を行い、記法の違いによるトレードオフを評価できる。
4	ソフトウェア設計で説明した設計手法の振り返り(その3)設計手法(データ構造を中心とした設計)について説明する。	DES.str.3 データ構造を中心とした設計	DES.str.3 データ中心による設計手法の概念(定義、適用条件、考え方、プロセス、主要成果物)について説明できる。基本的な要求に対して、定義されたプロセスに従って設計成果物を作成することができる。	DES.str.3 機能指向、オブジェクト指向、データ中心設計、アスペクト指向などの手法を適材適所で組み合わせ、要求定義から導出された仕様に基づきソフトウェアの設計ができる。設計結果に対して、適用した手法の妥当性を評価できる。
	アーキテクチャスタイル(その1)アーキテクチャスタイルの基本概念について説明する。	DES.ar.1 アーキテクチャスタイル(パイプアンドフィルタ、レイヤード、トランザクション中心、ピアツーピア、publish/subscribe、イベント駆動、クライアントサーバなど)	DES.ar.1 代表的なアーキテクチャスタイル(一般的構造、分散システム、対話型、順応型、その他)の基本概念(名称、性質、適用条件、構造、振舞い)を説明できる。	DES.ar.1 アーキテクチャスタイルの基本概念に基づいてソフトウェアを設計できる。適用したスタイルの妥当性を説明できる。

授業展開	学習成果	トピック	学習目標	発展学習目標
5	アーキテクチャスタイル(その2) アーキテクチャスタイルの続きと、アーキテクチャで考慮すべき各種特性について説明する。  演習では、事例に基づき、アーキテクチャスタイルの具体的な適用方法、各種特性のトレードオフについて議論し、結果を発表する。	DES.ar.1 アーキテクチャスタイル(パイプアンドフィルタ、レイヤード、トランザクション中心、ピアツーピア、publish/subscribe、イベント駆動、クライアントサーバなど)	DES.ar.1 代表的なアーキテクチャスタイル(一般的構造、分散システム、対話型、順応型、その他)の基本概念(名称、性質、適用条件、構造、振舞い)を説明できる。	DES.ar.1 アーキテクチャスタイルの基本概念に基づいてソフトウェアを設計できる。適用したスタイルの妥当性を説明できる。
	DES.ar.2 アーキテクチャで考慮すべき様々な特性間のトレードオフ	DES.ar.2 アーキテクチャで考慮すべき様々な特性(正当性、完全性、構築可能性等)の概念の定義、特性間の設計における影響を説明できる。	DES.ar.2 様々な特性を考慮したアーキテクチャを定義しソフトウェアを設計できる。設計した結果のトレードオフを説明できる。	
6	アーキテクチャで考慮すべき各種特性およびこれらの中でのトレードオフの前の続きを説明する。また、演習の続きを行う。	DES.ar.2 アーキテクチャで考慮すべき様々な特性間のトレードオフ	DES.ar.2 アーキテクチャで考慮すべき様々な特性(正当性、完全性、構築可能性等)の概念の定義、特性間の設計における影響を説明できる。	DES.ar.2 様々な特性を考慮したアーキテクチャを定義しソフトウェアを設計できる。設計した結果のトレードオフを説明できる。
7	ハードウェアに関連してソフトウェアアーキテクチャで考慮すべき事柄と、要求のトレーサビリティについて説明する。	DES.ar.3 ソフトウェアアーキテクチャで考慮すべきハードウェア	DES.ar.3 ソフトウェアアーキテクチャの定義上、ハードウェアによって影響を受ける事柄について(機能性、性能、使用性等)基本事項を説明できる。	DES.ar.3 ハードウェアを考慮したソフトウェアの設計ができる。設計した結果の妥当性を説明できる。
		DES.ar.4 アーキテクチャにおける要求のトレーサビリティ(その1) ソフトウェアアーキテクチャと要求	DES.ar.4 アーキテクチャ定義結果に対して要求とのトレーサビリティ(追跡可能性)の確保と検証のための基本概念(設計と要求の対応関係の必要性、マトリクスによる追跡方法)を説明できる。	DES.ar.4 要求とのトレーサビリティを考慮し、ソフトウェアを設計できる。設計結果に対して、要求を満足していることの妥当性の検証ができる。
8	最適化アーキテクチャ(その1) システム全体を考慮したアーキテクチャ設計の最適化について説明する。アーキテクチャ設計にかかわる要求および設計手法について説明する。	システム全体の最適化 アーキテクチャ設計にかかわる要求	- システム全体に対して、システム管理・運用、セキュリティ、ネットワーク、プラットフォーム(OS、ミドルウェアなど)、パフォーマンス、可用性、物理データ構造などに関する要求の考慮が必要であることが説明できる。	- システム全体の最適化を考慮した設計ができる。設計した結果の妥当性を説明できる。
		システム全体の最適化 アーキテクチャ設計手法	- 上記要求に対する典型的な設計方法と妥当性の根拠についてを理解し、システム全体で最適化されたアーキテクチャ設計とは何かを説明できる。	-
9	最適化アーキテクチャ(その2) システム全体を考慮したアーキテクチャ設計の最適化についての前の続きを説明する。アーキテクチャ設計の評価手法について説明する。	システム全体の最適化 アーキテクチャ設計手法	- システム全体最適化要求に対する典型的な設計方法と妥当性の根拠についてを理解し、システム全体で最適化されたアーキテクチャ設計とは何かを説明できる。	- システム全体の最適化を考慮した設計かどうか、実現可能性を検証できる。
		システム全体の最適化 アーキテクチャ設計評価手法	- システム全体で最適化されたアーキテクチャ設計に対して、実現可能性を評価する必要があることならびに、代表的な評価観点や評価方法を説明できる。	-
10	大規模システムの開発の場合のアーキテクチャ設計手法について、手法の全体像、大規模システムに対する主な要求、その要求に対応するためのアーキテクチャ設計の考え方について説明する。	手法の全体像	- ソフトウェア/システムアーキテクチャのパターン、設計手法、特性、適用条件、メリット、デメリット等の全体像を説明できる。	- 大規模システムの最適化を考慮した設計ができる。設計した結果の妥当性を説明できる。
		大規模システムに対する要求	- 大規模システムの開発では、複数のサブシステムを組み合わせた設計になることを理解し、典型的な大規模システムへの信頼性、拡張性、セキュリティ対策も含めた要求を説明できる。	-
		大規模システムに対するアーキテクチャ設計の考え方	- 大規模システムへの要求に対して、アーキテクチャ設計の手法を適切に組み合わせることが必要になることを理解し、典型的な組合せについて説明できる。	-
11	システム全体のアーキテクチャと要求のトレーサビリティについてと、ドメイン特化のアーキテクチャおよびプロダクトラインについて説明する。	DES.ar.4 アーキテクチャにおける要求のトレーサビリティ(その2) システム全体のアーキテクチャと要求	DES.ar.4 アーキテクチャ定義結果に対して要求とのトレーサビリティ(追跡可能性)の確保と検証のための基本概念(設計と要求の対応関係の必要性、マトリクスによる追跡方法)を説明できる。	DES.ar.4 要求とのトレーサビリティを考慮し、ソフトウェアを設計できる。設計結果に対して、要求を満足していることの妥当性の検証ができる。 必要に応じ、グループ演習を通して、要求を満足していることの妥当性検証を確認する。
		DES.ar.5 ドメイン特化アーキテクチャおよびプロダクトライン開発	DES.ar.5 ドメイン特化型開発の定義、プロダクトライン型開発の考え方、アーキテクチャが考慮すべき事柄、アーキテクチャ定義の重要性が説明できる。	DES.ar.5 ドメイン特化の事例を想定してソフトウェアを設計できる。設計結果について、妥当性を評価できる。
12	設計上の特性とそのマトリクス(その1) 設計上の特性の測定方法(定量的方法、定性的方法)と評価方法について説明する。	DES.ste.2 設計上の特性の測定(結合度、凝集度、情報隠蔽、関心事の分離など)	DES.ste.2 設計上の特性(結合度、凝集度、情報隠蔽、関心事の分離)について、その測定方法(測定のねらい、特性を測る方法、測定値の判断方法、判断根拠)について説明できる。	DES.ste.2 ソフトウェアの設計結果に対して、特性の測定ができる。測定結果に基づいて、ソフトウェアの設計の評価ができる。
		アーキテクチャの定性的な評価	アーキテクチャを定性的に評価する手法(評価のねらい、評価手順、評価対象、判断根拠)について説明できる。	
		DES.ste.3 設計のマトリクス(アーキテクチャ上の要因、変換、一般的な使い方におけるマトリクス)	DES.ste.3 設計の良し悪しを判定するマトリクス(種類、役割、使い方、計測方法、解釈方法)について説明できる。	DES.ste.3 ソフトウェアの設計結果に対して、代表的な設計のマトリクスを計測できる。計測結果に基づいて、ソフトウェアの設計が評価できる。
13	設計上の特性とそのマトリクス(その2) 設計上の特性の測定方法と評価方法(前回の続き)、アーキテクチャの記述方法とツールサポートについて説明する。	DES.ste.3 設計のマトリクス(アーキテクチャ上の要因、変換、一般的な使い方におけるマトリクス)	DES.ste.3 設計の良し悪しを判定するマトリクス(種類、役割、使い方、計測方法、解釈方法)について説明できる。	DES.ste.3 ソフトウェアの設計結果に対して、代表的な設計のマトリクスを計測できる。計測結果に基づいて、ソフトウェアの設計が評価できる。
		DES.ste.4 アーキテクチャの形式的な記述と設計の分析	DES.ste.4 アーキテクチャを形式的に記述するための基本概念(設計の定義、役割、特性、メリット、限界、ADL: Architecture Description Language)を説明できる。	DES.ste.4 アーキテクチャを形式的に記述し、ソフトウェアの設計ができる。設計結果に対して、形式的に記述する場合としない場合のトレードオフを評価できる。
		DES.ar.6 アーキテクチャのための記法(アーキテクチャ上のビューポイントと表現、コンポーネント図など)	DES.ar.6 アーキテクチャを記述するための代表的なADL(名称、役割、読み方、書き方)が説明できる。構造と振舞いに関して、1つ以上の記法によって、設計結果を記述できる。	DES.ar.6 代表的な記法を利用したアーキテクチャの設計ができる。複数の記法によって、同一の要求の設計を行い、記法の違いによるトレードオフを評価できる。
		DES.ste.1 設計支援ツール(アーキテクチャ、静的解析、動的評価など)	DES.ste.1 設計、設計の検証を支援する代表的なツール(名称、役割、使い方、特性)を説明できる。	DES.ste.1 設計支援ツールを利用したソフトウェアの設計ができる。
14	重要事項のまとめ (各回に時間を分配し、当日の概要及び手法の全体像等について説明することも可)			
15	期末試験			
教科書・参考書	①岸、野田、深澤著:「ソフトウェアアーキテクチャ」ソフトウェアテクノロジーシリーズ4、アーキテクチャとドメイン指向トラック、共立出版 ②Len Bass, Rick Kazman, Paul Clements 著:Software Architecture in Practice (SEI Series in Software Engineering), Addison-Wesley Pub. 翻訳版「実践ソフトウェアアーキテクチャ」日刊工業新聞社 (※必要に応じ、機能指向、オブジェクト指向、アスペクト指向に関する設計手法の文献も選択する。)			
備考				

授業科目名	ソフトウェア設計
単位数	2
開設学期	2年生後期
目的	ソフトウェア設計の概念及び、開発ライフサイクル内での位置づけを学ぶとともに、いくつかの具体的な設計手法に関して学ぶことを目的とする
概要	ソフトウェア設計の概念を学んだ後、詳細設計の技法として、機能指向の設計、データ構造を中心とした設計、オブジェクト指向の設計、コンポーネント設計に関して学ぶ。
目標	ソフトウェア設計の概念と設計時に考慮すべき視点を理解し、それぞれの代表的な設計方法、表記方法を身に付けることで、様々なソフトウェア開発に対応できる能力を身につける。
先修科目	
関連科目	
授業方法	講義、机上演習 演習は設計支援ツールを用いる
評価方法・基準	演習課題および、最終試験を総合的に判断する。

授業展開	学習成果	トピック	学習目標	発展学習目標	
1	設計の概念、基本的な考慮事項、開発ライフサイクルにおける位置づけ、を説明する。	DES.con.1	設計という概念の定義	DES.con.1	DES.con.1
		DES.con.3	複数のソフトウェア開発ライフサイクルにおける設計の関係	DES.con.3	DES.con.3
		DES.con.4	設計の原則	DES.con.4	DES.con.4
		DES.con.2	基本的な設計の考慮事項	DES.con.2	DES.con.2
2	構造化分析設計法を通して、機能指向の設計に関して説明する。	DES.str.1	機能指向による設計	DES.str.1	DES.str.1
		DES.dd.1	詳細設計の技法(SSA/SD)	DES.dd.1	DES.dd.1
		DES.dd.5	データフロー図	DES.dd.5	DES.dd.5
		DES.dd.1	詳細設計の技法(SSA/SD)	DES.dd.1	DES.dd.1
3	構造化設計法と凝集度、結合度によるモジュール評価法を説明する。	DES.str.1	機能指向による設計	DES.str.1	DES.str.1
		DES.dd.1	詳細設計の技法(SSA/SD): STS分割、TR分割	DES.dd.1	DES.dd.1
		DES.dd.5	構造図	DES.dd.5	DES.dd.5
		DES.con.4	設計の原則	DES.con.4	DES.con.4
4	データ中心設計に関して説明する	DES.str.3	データ構造を中心とした設計	DES.str.3	DES.str.3
		DES.dd.5	ER図	DES.dd.5	DES.dd.5
		DES.dd.1	詳細設計の技法(JSD)	DES.dd.1	DES.dd.1
		DES.str.2	オブジェクト指向による設計	DES.str.2	DES.str.2
5	オブジェクト指向設計法の導入を行う。ユースケースとロバストネス分析、CRC法を説明する	DES.str.2	オブジェクト指向による設計	DES.str.2	DES.str.2
		DES.str.2	ユースケースとロバストネス分析	DES.str.2	DES.str.2
		DES.str.2	CRC法	DES.str.2	DES.str.2
		DES.str.2	オブジェクト指向による設計	DES.str.2	DES.str.2
6	オブジェクト指向における基本的な概念を説明する UMLとUMLエディタを利用して、静的構造の基本的な表現方法を説明する 例題を利用した、基本的静的構造のUMLモデリング演習を行う	DES.str.2	オブジェクト指向による設計	DES.str.2	DES.str.2
	UMLによる、設計の静的構造の表現方法を説明する	DES.dd.5	UML (静的構造図)	DES.dd.5	DES.dd.5
		DES.ste.1	設計支援ツール	DES.ste.1	DES.ste.1
		DES.dd.5	UML (静的構造図)	DES.dd.5	DES.dd.5
7	UMLによる、設計の動的構造の表現方法を説明する	DES.dd.5	UML (静的構造図)	DES.dd.5	DES.dd.5
		DES.dd.5	UML (相互作用図、シーケンス、アクティビティ)	DES.dd.5	DES.dd.5
		DES.dd.5	UML (相互作用図、シーケンス、アクティビティ)	DES.dd.5	DES.dd.5
		DES.dd.5	UML (相互作用図、状態機械図、その他)	DES.dd.5	DES.dd.5
8	状態機械図と他の相互作用図の説明を行う 状態機械図による設計表現方法を説明する リアクティブシステムの設計方法に関して説明する	DES.dd.5	UML (相互作用図、状態機械図、その他)	DES.dd.5	DES.dd.5
		DES.dd.1	状態機械図を用いた設計	DES.dd.1	DES.dd.1
		DES.dd.5	UML (相互作用図、状態機械図、その他)	DES.dd.5	DES.dd.5
		DES.con.4	設計の原則	DES.con.4	DES.con.4
9	契約による設計法を通して、オブジェクト指向設計における設計原則を説明する	DES.con.4	設計の原則	DES.con.4	DES.con.4
	コンポーネントとシステムのインタフェース設計を説明し、契約による設計法を説明する	DES.dd.3, DES.dd.4	コンポーネント設計	DES.dd.3, DES.dd.4	DES.dd.3, DES.dd.4
		DES.dd.1	詳細設計の技法(DbC)	DES.dd.1	DES.dd.1
		DES.dd.3	コンポーネント指向開発、DCOM, EJB	DES.dd.3	DES.dd.3
10	既存のコンポーネント技術に関して説明を行う	DES.dd.3	コンポーネント指向開発、DCOM, EJB	DES.dd.3	DES.dd.3
		DES.dd.2	デザインパターンの概要、Adapter, Decorator, Composite, Factory, Object Pool	DES.dd.2	DES.dd.2
11	構造、生成に関するデザインパターンを説明する	DES.dd.2	デザインパターンの概要、Adapter, Decorator, Composite, Factory, Object Pool	DES.dd.2	DES.dd.2
		DES.dd.5	パターン適用済みコードの設計復元演習	DES.dd.5	DES.dd.5

授業展開	学習成果	トピック	学習目標	発展学習目標
12	振る舞いに関するデザインパターンを説明する	DES.dd.2 Mediator, Façade, Command Observer, State, Strategy	DES.dd.2 振る舞いに関するデザインパターンを知っている	DES.dd.2 振る舞いに関するデザインパターンを利用できる
		DES.dd.5 パターン適用済みコードの設計復元演習	DES.dd.5	DES.dd.5
13	フレームワークに関して説明する	DES.dd.3 フレームワークの概念、制御の反転、既存のフレームワークの説明	DES.dd.3 フレームワークの概念を理解している	DES.dd.3 制御の反転、依存注入に関して説明できる
14	重要事項のまとめ			
15	期末試験			
教科書・参考書				
備考				

授業科目名	<b>検証と妥当性確認(V&amp;V)</b>	
単位数	2	
開設学期	3年生後期	
目的	検証と妥当性確認について、その目的と役割を理解し、基本的な事項を説明できることを目的とする。	
概要	検証と妥当性確認では、プログラムが仕様を満足していることや、ステークホルダーの期待を実現していることを保証するために、システムを検査する静的あるいは動的技法の両方を使う。静的技法は、ソフトウェアのライフサイクルのすべての段階を通じて、システム内部表現の解析や検査に用いる。一方、動的技法は、実装されたシステムにのみ適用できる。	
目標	検証と妥当性確認で用いられる用語について、また、各種レビューおよびテスト技法の目的や役割について、それぞれ理解するとともに、検証と妥当性確認の活動を実施するための基礎的事項を習得する。	
先修科目		
関連科目		
授業方法		
評価方法・基準		
授業展開		

1	学習成果	トピック	学習目標	発展学習目標
1	授業の概要と目的および進め方について説明する(※講義0.3h)。V&Vの概念、基本的な考慮事項、計画や戦略について説明する。	VAV.fnd.1 V&Vの目的と制約	VAV.fnd.1 V&Vの目的と制約について理解し、説明できる。	
		VAV.fnd.2 V&Vの計画	VAV.fnd.2 V&Vの計画の目的を理解し、説明できる。	VAV.fnd.2 V&Vの計画として、資源、役割、および責任が、そのそれぞれに対して明確に割り当てられていることを確認できる。
		VAV.fnd.3 V&Vの戦略のドキュメント化(テストなど)	VAV.fnd.3 V&Vの戦略のドキュメント化(テストなど)の目的や役割を理解し、説明できる。	VAV.fnd.3 V&Vの戦略の実際のドキュメント化として、さまざまな資源、それらの役割およびアクティビティが、利用されるツールと共に記載することができる。
2	V&Vの開発ライフサイクルにおける位置づけや、要求や品質との関係について説明する。	VAV.fnd.4 メトリクスと測定(信頼性、ユーザビリティ、性能など) VAV.fnd.5 V&Vに関連する活動	VAV.fnd.4 各種メトリクスについて、その目的や役割を理解し、説明できる。 VAV.fnd.5 V&Vに関連する活動(要求分析、形式手法、保守、ソフトウェア品質など)について、その関連性を理解し、説明できる。	VAV.fnd.4 各種メトリクスを実際に使い、利用できる。
3	レビューについて説明する。	VAV.rev.1 机上チェック VAV.rev.2 ウォークスルー VAV.rev.3 インスペクション	VAV.rev.1 机上チェックの役割を理解し、説明できる。 VAV.rev.2 ウォークスルーの役割を理解し、説明できる。 VAV.rev.3 インスペクションの役割を理解し、説明できる。	VAV.rev.1 机上チェックを実際に使い、実施できる。 VAV.rev.2 ウォークスルーを実際に使い、実施できる。 VAV.rev.3 インスペクションを実際に使い、実施できる。
4	単体テストについて説明する。	VAV.tst.1 単体テスト VAV.tst.2 例外のハンドリング(例外のハンドリングを発生させるテストの設計、よい例外ハンドリングの設計)	VAV.tst.1 単体テストの目的や役割を理解し、説明できる。 VAV.tst.2 例外のハンドリングを発生させるテストの設計や、よい例外ハンドリングの設計の重要性を理解し、説明できる。	
5	ホワイトボックステスト技法について説明する。	VAV.tst.3 網羅性分析(命令網羅、分岐網羅、基本パス網羅、複合条件網羅、データフローパス網羅など)	VAV.tst.3 網羅性分析(命令網羅、分岐網羅、基本パス網羅、複合条件網羅、データフローパス網羅など)の役割を理解し、説明できる。	VAV.tst.3 網羅性分析(命令網羅、分岐網羅、基本パス網羅、複合条件網羅、データフローパス網羅など)を実際に使い、実施できる。
6	ブラックボックステスト技法について説明する。(1)	VAV.tst.4 ブラックボックスの技法	VAV.tst.4 ブラックボックステスト技法(同値分割、境界値分析、デシジョンテーブルなど)の役割を理解し、説明できる。	VAV.tst.4 ブラックボックステスト技法(同値分割、境界値分析、デシジョンテーブルなど)を実際に使い、実施できる。
7	ブラックボックステスト技法について説明する。(2)	VAV.tst.4 ブラックボックスの技法	VAV.tst.4 ブラックボックステスト技法(状態遷移テスト、ペア構成テストなど)の役割を理解し、説明できる。	VAV.tst.4 ブラックボックステスト技法(状態遷移テスト、ペア構成テストなど)を実際に使い、実施できる。
8	ソフトウェア結合テストについて説明する。	VAV.tst.5 ソフトウェア結合テスト	VAV.tst.5 ソフトウェア結合テストの目的や役割を理解し、説明できる。	VAV.tst.5 ドライバやスタブの役割について理解した上で、実際に使うことができる。
9	ユースケーステスト技法や運用プロファイルによるテストについて説明する。	VAV.tst.6 ユースケースや顧客シナリオによるテストケースの設計 VAV.tst.7 運用プロファイルによるテスト	VAV.tst.6 ユースケーステスト技法の役割を理解し、説明できる。 VAV.tst.7 運用プロファイルによるテストの目的や役割を理解し、説明できる。	VAV.tst.6 ユースケーステスト技法を実際に使い、実施できる。
10	システム総合テストや品質特性に関連するテストについて説明する。	VAV.tst.8 システム総合テストと受入れテスト VAV.tst.9 品質特性に関連するテスト(ユーザビリティ、セキュリティ、互換性、アクセシビリティなど)	VAV.tst.8 システム総合テストと受入れテストの目的や役割を理解し、説明できる。 VAV.tst.9 品質特性に関連するテスト(ユーザビリティ、セキュリティ、互換性、アクセシビリティなどの非機能的な要求)の目的や役割を理解し、説明できる。	
11	回帰テストやテストツールについて説明する。	VAV.tst.10 回帰テスト VAV.tst.11 テストツール	VAV.tst.10 回帰テストの目的や役割を理解し、説明できる。 VAV.tst.11 各種テスト支援ツールについて、名称、役割、特性を理解し、説明できる。	VAV.tst.11 テスト支援ツールを実際に使い、テストを実施できる。
12	不具合分析について説明する。(1)	VAV.par.1 不具合報告書の分析 VAV.par.2 デバッグや不具合切り分けの技法	VAV.par.1 不具合報告書の役割を理解し、説明できる。 VAV.par.2 デバッグや不具合切り分けの技法について理解し、説明できる。	VAV.par.1 不具合報告書を分析するとともに、実際に、不具合報告書を書くことができる。 VAV.par.2 デバッグを実際に使い、デバッグを実施できる。
13	不具合分析について説明する。(2)	VAV.par.3 欠陥分析 VAV.par.4 不具合の追跡 不具合管理	VAV.par.3 欠陥分析の目的や役割を理解し、説明できる。 VAV.par.4 不具合追跡の目的や役割を理解し、説明できる。 不具合管理の役割を理解し、説明できる。	VAV.par.3 欠陥に対する分析を行い、いつそれらがソフトウェアに入り込んだか、どのようなエラーがこれを生じる原因になったか(曖昧に定義された要求、不正な変数宣言、メモリリーク、プログラミングでの構文誤りなど)、いつそれが発見されたかなどを決定できる。 VAV.par.4 不具合追跡情報を、どの局面での改善が必要か、および過去の分析やテストがどのような効果を挙げたかを決定するために利用できる。 バグトラッキングシステム(Bug Tracking System: BTS)の役割を理解し、実際に使うことができる。
14	重要事項のまとめ			
15	期末試験			

授業展開	学習成果	トピック	学習目標	発展学習目標
教科書・参考書	「はじめて学ぶソフトウェアのテスト技法」リー・コーブランド 著, 宗雅彦 訳 (日経BP社) ISBN 4-8222-8251-1 「JSTQB認定テスト技術者 Foundation Level試験」大西建児, 勝亦匡秀, 加藤大受, 佐々木方規, 鈴木三紀夫, 町田欣史, 吉澤智美, 湯本剛 著 (翔泳社) ISBN: 978-4-7981-1204-6			
備考				



授業科目名	<b>形式手法</b>					
単位数	2					
開設学期	3年生前期					
目的	形式手法に関する基礎知識、様々な種類の仕様記述言語、検証方法などについて学ぶ。					
概要	様々な仕様記述言語の概要から、いくつかの仕様記述言語を選択し、その詳細を演習を通じて学ぶ。					
目標	形式手法に関する基礎的な知識を獲得することを目標とする。					
先修科目	離散数学、論理と推論・計算論理					
関連科目						
授業方法	座学と演習					
評価方法・基準	期末試験、または、必要に応じて中間試験の結果により評価を行う。					
授業展開		学習成果	トピック	学習目標		発展学習目標
1	授業の概要と目的および進め方について説明する(※講義0.3h)。モデリングの原則について説明をする	MAA.md.1	モデリングの原則(抽象化、汎化)	MAA.md.1		MAA.md.1 抽象化に関連する諸概念やモデリングに関する広範囲で深い理解を得る。
	モデリング言語一般について説明をする	MAA.md.5	非形式的モデリング言語(例、UML)と形式仕様記述言語の特徴、違い	MAA.md.5	形式的、非形式的モデリング言語の設計原理、言語の特徴、モデリングの方法の特徴を理解する。	MAA.md.5
2	形式仕様記述言語全般について概説する。	MAA.md.4, MAA.af.6	形式仕様記述言語(モデルベース、代数型、プロセス代数の概略的な紹介。	MAA.md.4	モデルベース仕様記述言語、代数型仕様記述言語、プロセス代数、抽象機械表記法などの仕様記述言語としての違いを理解する。	MAA.md.4 様々な特徴を持った仕様記述言語がどのようなシステムの仕様記述に適しているかが理解できるようになる。
3	抽象機械について概説する。	MAA.md.3	抽象機械、構文、仕様記述の見本、様々な抽象機械の紹介	CMP.fm.1	言語の定義、基本的な例から、モデル化の方法を学ぶ。	CMP.fm.1 異なる抽象機械(SDL, asml等)に対して、その違いが理解できるようになる。
4	抽象機械を用いて、簡単な例を記述し、演習を行う。	CMP.fm.1, MAA.md.5	例題から演習問題を解く。	CMP.fm.1	抽象機械を用いて、どのようにシステムのモデル化に利用することが出来るかを理解する。	CMP.fm.1
5	抽象機械を用いて、より高度な演習を行う。	CMP.fm.1, MAA.md.5	高度な演習問題を解く。	CMP.fm.1	抽象機械による、高度な演習問題が解けるようになる。	CMP.fm.1
6	形式仕様記述言語(例えばZ)について概要を説明する	MAA.md.3	仕様記述言語(例えばZ)、構文、意味論、仕様記述の見本	MAA.md.3	言語の定義、意味論、基礎法則と、基本的な例から、仕様記述(モデル化)方法の特徴について学ぶ。	MAA.md.3 異なる形式仕様記述言語に対して、その違いが理解できるようになる。
7	形式仕様記述言語(Z、B、VDM)を用いて、簡単な例を記述し、演習を行う。	CMP.fm.2, MAA.md.5	例題から演習問題を解く。(例えばZ)、関係、関数、列による仕様記述。	CMP.fm.2	形式仕様記述言語(Z、B、VDM)による簡単な仕様記述方法を理解する。	CMP.fm.2
8	形式仕様記述言語(Z、B、VDM)を用いて、より高度な演習を行う。	CMP.fm.2, MAA.md.5	高度な演習問題を解く。(例えばZ)、プロモーション。	CMP.fm.2	形式仕様記述言語(Z、B、VDM)による高度な演習問題(例えば、プロモーション)が出来るようになる。	CMP.fm.2
9	記述の正さと、意味の正しさの違いについて、正当性と妥当性の違いについて説明をする。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	形式仕様における記述と意味の正しさ。正当性、妥当性の概念。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	記述と意味の違い、および、正当性と妥当性の違いを説明できるようになる。仕様とプログラムの関係を理解する。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6 実際のソフトウェアに対して正当性と妥当性の違いが理解でき、それらにより保証されるソフトウェアの信頼性を検討できるようになる。
10	検証対象のプログラムと検証する性質について説明をする。仕様と記述する性質の関係、記述の明快性を説明する。	MAA.md.2, MAA.md.6, CMP.fm.4, CMP.fm.5, CMP.fm.6	プログラムの構造(WHILEプログラム、フローチャート)。事前条件、事後条件、不変表明。	MAA.md.2, CMP.fm.4, CMP.fm.5, CMP.fm.6	仕様と事前条件、事後条件、不変表明の関係、および、検証対象のプログラムの振る舞いと、事前条件、事後条件、不変表明の関係を理解する。	MAA.md.2, CMP.fm.4, CMP.fm.5, CMP.fm.6 仕様に基づいてソフトウェアの事前条件、事後条件、不変表明を適切に記述できるようになる。
11	代表的なプログラムの検証法。保証できる性質について説明をする。仕様を満たすプログラムについて説明する。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	代表的なプログラムの検証法(Floyd, Hoareなど)。保証できる性質(部分正当性、停止性、全正当性)。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	プログラムの検証法を理解する。仕様を満たすプログラムがどのようなものであるか理解する。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6 ソフトウェアが仕様を満たしていることや期待した性質を持つことを保証する方法を検討できるようになる。プログラム意味論について理解する。コンピュータによる検証支援(検証条件生成器、定理証明やモデル検査)について理解する。
12	簡単な例の検証を行うことで、学生の理解を深める。仕様を満たすプログラムの候補について検討する。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	乗算・除算、階乗、べき乗などのプログラムの検証	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6	具体的なプログラムを検証できるようになる。仕様を満たすプログラムの候補を検討できるようになる。	CMP.fm.8, CMP.fm.4, CMP.fm.5, CMP.fm.6 ソフトウェアが仕様を満たしていることや期待した性質を持つことを検証できるようになる。
13	仕様から実装を行う手法について説明をする。プログラムを自動生成したり、仕様から詳細化することによりプログラムを導出するという概念を説明する。	CMP.fm.3, CMP.fm.4, CMP.fm.5, CMP.fm.6, CMP.fm.7	仕様から実装へ変換する手法。	CMP.fm.3, CMP.fm.4, CMP.fm.5, CMP.fm.6, CMP.fm.7	仕様から正しいプログラムを作成するという概念を理解する。詳細化という概念を理解する。仕様からある程度のプログラムを自動生成可能であることを理解する。	CMP.fm.3, CMP.fm.4, CMP.fm.5, CMP.fm.6, CMP.fm.7 仕様から正しい実装を形式的に導けるようになる。仕様からプログラムを自動生成する方法を検討できるようになる。詳細化計算について理解する。
14	重要事項のまとめ					
15	期末試験					
教科書・参考書	J. Woodcock, J. Davies, "Using Z", Prentice Hall (1996), 荒木啓二郎 張漢明, 「プログラム仕様記述論」, オーム社(2002) Z. Manna: Mathematical Theory of Computation, McGraw-Hill( 1974). E.W.Dijkstra: A discipline of programming, Prentice-Hall(1976). 林晋: プログラム検証論, 共立出版(1995).					
備考						

授業科目名		ソフトウェアプロセスと品質			
単位数		2			
開設学期		3年生前期			
目的		ライフサイクル全体を通じたソフトウェアプロダクトの価値創出のための活動と技術について理解することを目的とする			
概要		ソフトウェアライフサイクルの主要工程、進化と保守、ソフトウェア品質の概念と管理手法、および、ソフトウェアプロセスの概念とその改善手法を学ぶ。			
目標		ソフトウェアライフサイクルの主要工程とその内容を説明できる ソフトウェア進化と保守の目的、作業、技術について説明できる ソフトウェア品質の概念や、その管理手法を説明できる ソフトウェアプロセスの概念とその改善手法を説明できる			
先修科目					
関連科目					
授業方法		講義			
評価方法・基準		下記により評価する。 ・中間および期末試験の結果			
授業展開		学習成果	トピック	学習目標	
1	授業の概要と目的および進め方について説明する。 ソフトウェアライフサイクルをシステム価値創出のためのプロセスとして理解し、価値実現のための技術として品質、プロセス、保守といった概念を位置づけて説明する	FND.ec.1、 PRO.imp.7	ソフトウェアライフサイクルを通じた価値の考慮	FND.ec.1、 PRO.imp.7	ISO12207などを題材に、企画から開発、保守、進化、廃棄といったライフサイクルについて理解する
		FND.ec.2	システム目的の策定	FND.ec.2	ユーザ参加型設計、ステークホルダー間のWin-Win関係、品質機能展開、プロトタイプピングなどの概念を理解する
		FND.ec.3、 PRF.pr.7	費用対効果の評価、雇用形態	FND.ec.3、 PRF.pr.7	利益実現、トレードオフ分析、コスト分析、ROI分析などの経済的な視点を知る。代表的な雇用形態について理解し、説明できる
		FND.ec.4	システム価値の実現	FND.ec.4	優先順位付け、リスクの解決、コストのコントロールなど
2	ライフサイクルにおける進化や保守の位置づけ	EVO.pro.1	進化や保守の基礎	EVO.pro.1	ライフサイクルにおける保守、進化の位置づけについて理解する
		EVO.pro.2	進化や保守の対象間関係	EVO.pro.2	進化・保守のための前提や、要求、アーキテクチャ、設計、ソースコードなど
		EVO.pro.3	進化や保守のモデル	EVO.pro.3	理論面、法制面などの観点から進化や保守プロセスについて理解する
		EVO.pro.4	進化や保守のコストモデル	EVO.pro.4	経済的観点から進化や保守プロセスについて理解する
		EVO.pro.5	進化や保守の計画	EVO.pro.5	アウトソーシング、内製など
3	進化や保守のための技術1	EVO.ac.1	レガシーシステムの扱い	EVO.ac.1	レガシーシステム、ラッパーなどの概念を理解する
		EVO.ac.2	プログラムの理解とリバースエンジニアリング	EVO.ac.2	進化・保守におけるプログラム理解とリバースエンジニアリングの重要性を理解する
		EVO.ac.3	システムおよび業務プロセスのリエンジニアリング	EVO.ac.3	(技術面とビジネス面での)システムおよび業務プロセスのリエンジニアリングの概念を理解する
4	進化や保守のための技術2	EVO.ac.4	影響解析	EVO.ac.4	保守における影響解析の重要性を理解し、クローン検出など代表的な手法を知る
		EVO.ac.5	マイグレーション(移行)	EVO.ac.5	(技術面とビジネス面での)マイグレーション(移行)
		EVO.ac.6	リファクタリング	EVO.ac.6	リファクタリングの概念を理解し、代表的な手法を知る
		EVO.ac.7	プログラム変換	EVO.ac.7	プログラム変換の概念を理解し、代表的な手法を知る
		EVO.ac.8	データのリバースエンジニアリング	EVO.ac.8	保守におけるデータのリバースエンジニアリングの必要性と概念を理解する
5	ソフトウェア品質の概念	QUA.cc.1	品質という概念の定義	QUA.cc.1	品質という概念の定義を理解する
		QUA.cc.2	品質に対する社会の関わり方	QUA.cc.2	品質に対する社会の関わり方
		QUA.cc.3	低い品質によって発生するコストと影響	QUA.cc.3	低い品質によって発生するコストと影響
		QUA.cc.4	品質モデルのコスト	QUA.cc.4	品質モデルのコスト
6	プロダクト保証とプロセス保証	QUA.cc.6	品質向上に対する取り組みの次元	QUA.cc.6	品質向上に対する取り組みの次元として、プロダクト保証、プロセス保証などがあることを知る
		QUA.pda.1	プロダクト保証の起源、保証とV&Vの違い	QUA.pda.1	プロダクト保証の意味・起源と、ソフトウェアにおけるプロダクト保証の考え方、V&Vとの違いについて概要を知る
		QUA.pda.2		QUA.pda.2	
		QUA.pda.3	プロダクト品質モデル、根本原因分析と欠陥予防	QUA.pda.3	ソフトウェアの品質特性(信頼性、ユーザビリティなど)についてISO9126等を題材に学び、根本原因分析や結果如棒といった品質管理技法の考え方を知る
		QUA.pda.4		QUA.pda.4	
		QUA.pca.1	プロセス保証の起源 品質計画 プロセス保証の整理と報告 プロセス保証の技法	QUA.pca.1	プロセス保証の意味・起源と、ソフトウェアにおけるプロセス保証の考え方、技法の概要について知る
		QUA.pca.2		QUA.pca.2	
		QUA.pca.3		QUA.pca.3	
QUA.pca.4	QUA.pca.4				
7	ソフトウェアの品質モデルとメトリクス	QUA.pda.5	プロダクト品質のメトリクスと測定	QUA.pda.5	ISO15936等の測定モデルに基づいたプロダクト品質の測定について学ぶ
		QUA.pda.6、 QUA.cc.5、	ソフトウェアの品質特性とアセスメント、品質機能展開	QUA.pda.6 QUA.cc.5	ユーザビリティ、信頼性、アベイラビリティなどの品質特性についてISO9126などを用いて説明できるようになる
		QUA.std.1 QUA.std.4	品質関連の標準	QUA.std.1 QUA.std.4	ISO9000シリーズ、ISO9126、ISO15939等、品質関連の標準についてその目的と概要について学ぶ
		QUA.cc.7	人、プロセス、技法、ツール、技術の役割	QUA.cc.7	ソフトウェア価値を実現するにおいて、人、プロセス、技法、ツール、技術の焼き割について理解し、プロセスの重要性を理解する。
8	ソフトウェアプロセスと品質	PRO.con.1	プロセスの概念と用語	PRO.con.1	プロセス分野における基本用語について学ぶ
		PRO.con.2	プロセスのインフラストラクチャ(開発者、ツール、教育など)	PRO.con.2	プロセスを実際に動かせるためのインフラストラクチャとしての開発者、ツールの役割を理解し、教育の重要性について理解する。
		PRO.con.3	プロセスのモデリングと仕様化	PRO.con.3	プロセスをモデリングし仕様かするという概念を知り、その意義を理解する

授業展開	学習成果	トピック	学習目標
9	プロセスのモデリング	PRO.imp.1	プロセス定義のレベル(組織、プロジェクト、チーム、個人など)
		PRO.imp.2	ライフサイクルモデル
		PRO.imp.4	個人によるソフトウェアプロセス(モデル、定義、測定、分析、改善)
		PRO.imp.5	チームによるソフトウェアプロセス(モデル、定義、組織、測定、分析、改善)
10	ソフトウェアプロセス標準	PRO.imp.3	ライフサイクルモデルの標準(IEEE標準、ISO標準など)
		PRO.imp.7 QUA.std.2	ISO 12207, 共通フレーム2007
		QUA.std.3 PRO.imp.6	組織ごとの標準の適用 プロセスのテーラリング
11	プロセスの改善～概要	PRO.con.7	日本の品質管理とプロセス改善 プロセスモデルのモデリングと分析
		PRO.con.5	プロセスの改善(個人の改善、チームの改善)
		PRO.con.4	プロセスの測定と分析
12	プロセスの改善～技術	QUA.pro.1	ソフトウェアプロセス改善のモデルとメトリクス
		QUA.pro.2	プロセス改善モデルの持つ品質に関する側面
		PRO.con.6	品質の分析とコントロール(欠陥予防、レビュー、品質特性、根本原因分析など)
13	プロセスの改善～標準	QUA.pro.3	ISO 15504の概要
		QUA.pro.4	ISO 15504における品質関連の領域
		QUA.pro.5	ソフトウェアCMMおよびCMMIにおける品質関連の領域
		QUA.pro.6	ソフトウェア産業に対するデミング賞・マルコムボルドリッジ賞の基準
	QUA.pro.7	その他のプロセス(改善)モデル	
14	重要事項のまとめ (各回に時間を分配し、当日の概要及び手法の全体像等について説明することも可)		
15	期末試験		
教科書・参考	ソフトウェアプロセス, 井上, 飯田, 松本, 共立出版, 2000 CMMI標準教本～プロセス統合と成果物改善の指針, メアリー ベス 他, JASPIC CMMI V1.1 翻訳研究会 訳, 日経BP, 2005 共通フレーム2007—経営者、業務部門が参画するシステム開発および取引のために、情報処理推進機構ソフトウェア・エンジニアリングセンター編, オーム社, 2007 ソフトウェアでビジネスに勝つ, ワッツ・ハンフリー, 富野 寿 訳, 構造計画研究所, 2003 初めて学ぶソフトウェアメトリクス～プロジェクト見積りのためのデータの導き方, ローレンス・H・バトナム, ウェア・マイヤーズ, 山浦 恒央 訳, 日経BP, 2005 Software Metrics: A Rigorous and Practical Approach, Norman Fenton, Shari Lawrence Pfleeger, Course Technology Ptr (Sd); 2nd Ed., 1998. 実践的ソフトウェア測定, ジョン・マギヤリー他著, 古山, 富野訳, 構造計画研究所, 2004 リファクタリング—プログラムの体質改善テクニック, マーチン ファウラー, 児玉他訳, ピアソンエデュケーション, 2000.		
備考			

授業科目名	ヒューマンファクター
単位数	2
開設学期	3年後期
目的	ソフトウェアの使いやすさは、使用する人間に関する知見なくしては開発できない。本講義では、利用者である人間の特性に着目し、使いやすいユーザインタフェースとはどのようなものかを理解しユーザインタフェース設計技法を身に付けることを目標とする
概要	ヒューマンコンピュータインタフェースは、人間の使う装置やシステムと人間の間境界に位置し、利用者にとっての使い易さや使い難さ、あるいは誤り難さや誤り易さなどを定める重要な要素である。ヒューマンコンピュータインタフェースの特性は、本来人間の感覚・知覚・認知特性、さらに習熟度等とも密接に関係し、これらの要素に十分配慮して定められるべきものである。このようなヒューマンコンピュータインタフェースに関して、その基本理念、設計指針、最適化、評価法などの問題を考え、学習する。
目標	人間の感覚や知覚などの特性が理解できる 良いインタフェースとはどのようなものが理解できる 入出力インタフェースが理解できる ガイドラインとモデルを知り、インタフェース設計に適用できる インタフェースの評価手法が理解できる
先修科目	
関連科目	
授業方法	講義、演習 ・ユーザインタフェース設計演習を行う。
評価方法・基準	下記により評価する。 ・演習への取り組み姿勢と発表成果 ・期末試験の結果

授業展開	学習成果	トピック	学習目標
1	授業の概要と目的、進め方、評価基準について説明する(0.3h) ヒューマンファクターとは何か、ヒューマンファクターとソフトウェアの関係を説明する	CMP.cf.7, CMP.cf.8	ヒューマンファクターの定義
	CMP.cf.7	ユーザサイドのヒューマンファクターとソフトウェアの関係	CMP.cf.6, CMP.cf.7
	CMP.cf.7	ユーザサイドのヒューマンファクターの要素	CMP.cf.6
2	ヒューマンコンピュータインタフェースとは何か、ヒューマンコンピュータインタフェースの3つの側面とアプローチ方法について説明し、あわせてヒューマンコンピュータインタフェース設計の基本原則について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.1	ヒューマンコンピュータインタフェースの定義
	DES.hci.1	ヒューマンコンピュータインタラクショントラック設計	DES.hci.1
	DES.hci.1	ヒューマンコンピュータインタフェースのアプローチ方法	DES.hci.1
	DES.hci.1	ヒューマンコンピュータインタフェース設計の基本原則	DES.hci.1
3	ヒューマンコンピュータインタフェース設計で重要になるインテリジェント科学の基礎的事項について解説し、メンタルモデルとは何か、メンタルモデルに合致したインタフェース設計について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.9	人間の感覚と知覚
	DES.hci.9	人間の認知と理解	DES.hci.9
	DES.hci.9	メンタルモデル	DES.hci.9
	DES.hci.9	人間の特性を考慮したヒューマンコンピュータインタフェース設計	DES.hci.9
4	入力装置と表示装置のインタフェースについて説明し、操作性についても触れる。表示装置の特徴を明らかにし音声によるインタフェースについて説明を加える。最後に視線入力によるインタフェースについても紹介する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.8	入力装置
	DES.hci.8	表示装置	DES.hci.8
	DES.hci.8	音声による入出力	DES.hci.8
	DES.hci.8	視線による入力技術	DES.hci.8

授業展開	学習成果	トピック	学習目標
5 ヒューマンコンピュータインタフェースの使いやすさに影響を与える人間とコンピュータとの対話方法に焦点をあて、メッセージをわかりやすく伝えるための手段について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.2	対話型システムの設計原則	DES.hci.2 対話型システムとは何か、対話型システムの特徴と課題を理解し、それらを踏まえた設計原則が理解できる
	Des.hci.2	モードやナビゲーションの利用	Des.hci.2 対話型システムの構成要素であるモードやナビゲーションの特徴、利用局面、利点、欠点が理解できる
	DES.hci.5	コマンド言語によるインタラクション	DES.hci.5 対話型システムの構成要素であるコマンド言語の特徴、利用局面、利点、欠点が理解できる
	DES.hci.5	メニュー選択によるインタラクション	DES.hci.5 対話型システムの構成要素であるメニュー選択の特徴、利用局面、利点、欠点が理解できる
	DES.hci.5	直接操作によるインタラクション	DES.hci.5 対話型システムの構成要素である直接操作の特徴、利用局面、利点、欠点が理解できる
	DES.hci.4	コンピュータの応答時間	DES.hci.4 対話型システムにおける適切な応答時間と、応答時間がユーザに与える影響が理解できる
6 GUIにおける配慮事項を述べ、GUI設計の留意事項を解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.3	表示デバイス	DES.hci.3 表示デバイスの種類と特徴が理解できる
	DES.hci.3	GUIの基本概念	DES.hci.3 GUIとは何か、GUIの特徴と構成要素が理解できる
	DES.hci.3	ウィンドウ設計	DES.hci.3 ウィンドウ設計に必要な考え方を理解し、設計方針が立案できる
	DES.hci.3	情報の視覚化	DES.hci.3 情報の視覚化の必要性とその効果が理解できる
7 ソフトウェアの設計における人間工学的原則と認知科学的原則について述べ、ソフトウェア設計におけるヒューマンコンピュータインタフェースの構造的、機能的視点について解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.7	ソフトウェア設計における人間工学的原則	DES.hci.7 人間工学とは何か、人間工学におけるユーザビリティの考え方を踏まえ、ユーザビリティを考慮したヒューマンコンピュータインタフェース設計の必要性が理解できる
	DES.hci.7	ソフトウェア設計における認知科学的原則	DES.hci.7 認知科学とは何か、認知科学におけるユーザビリティの考え方を踏まえ、ユーザビリティを考慮したヒューマンコンピュータインタフェース設計の必要性が理解できる
	DES.hci.7	ソフトウェア工学からのヒューマンコンピュータインタフェースへのアプローチ	DES.hci.7 ソフトウェア設計(開発)におけるヒューマンコンピュータインタフェース設計の位置づけが理解できる
	DES.hci.7	人間中心設計	DES.hci.7 人間中心設計の進め方が理解できる
8 ヒューマンコンピュータインタフェース設計の評価法について解説する	DES.hci.7, VAV.hct.1	有用性とユーザビリティの多面性	DES.hci.7, VAV.hct.1 有用性とユーザビリティに関する評価方法を理解し説明できる
	DES.hci.7, VAV.hct.2	経験則による評価	DES.hci.7, VAV.hct.2 経験則による評価について、その役割を把握し説明できる
	DES.hci.7, VAV.hct.3	認知的ウォークスルー	DES.hci.7, VAV.hct.3 認知的ウォークスルーについて、その役割を把握し説明できる
	DES.hci.7, VAV.hct.4	ユーザによるテストのためのアプローチ	DES.hci.7, VAV.hct.4 ユーザによるテスト(据付テスト、アルファ及びベータテスト)の目的や役割を把握し、説明できる
	DES.hci.7, VAV.hct.5	WebサイトやWebアプリケーションにおけるユーザビリティおよびテスト技法	DES.hci.7, VAV.hct.5 WebサイトやWebアプリケーションにおけるテスト技法の目的や役割を把握し、説明できる
9 ヒューマンエラーとは何かについて解説し、ヒューマンコンピュータインタフェース設計におけるヒューマンエラーの問題について考え、ヒューマンエラー防止に繋がるインタフェース設計について述べる(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.10	ヒューマンエラー	DES.hci.10 ヒューマンエラーとは何かが理解できる
	DES.hci.10	ヒューマンエラーのモデル	DES.hci.10 人間の情報を認知する段階に関連づけたヒューマンエラーモデルが理解できる
	DES.hci.10	エラーの原因、測定	DES.hci.10 エラーの原因、測定方法が理解できる
	DES.hci.10	エラー防止のためのヒューマンコンピュータインタフェース設計	DES.hci.10 ヒューマンエラー防止につながるインタフェース設計として、セル日作業環境要因に基づく設計、人間工学に基づく設計、認知工学に基づく設計、疲労、ストレス、負担などの観点に基づく設計指針が理解できる
10 インタフェースの国際的な標準を紹介した後、開発者サイドのヒューマンファクターについて解説する(冒頭0.1hで前回復習と今回講義の概要)	DES.hci.6	インタフェースと標準化	DES.hci.6 インタフェースの国際標準の動向、アイコンの国際標準などが理解できる
	CMP.cf.8	開発者サイドのヒューマンファクターとソフトウェアの関係	CMP.cf.6, CMP.cf.7 ソフトウェアに影響を与える開発者サイドのヒューマンファクターとは何かが説明できる
	CMP.cf.8	開発者サイドのヒューマンファクターの要素	CMP.cf.7 開発者サイドのヒューマンファクター(コメント、構造、仕様書やモデル、コードの可読性)とは何かが説明できる

授業展開	学習成果	トピック	学習目標	
11	開発者サイドのヒューマンファクターの解説を続ける(冒頭0.1hで前回復習と今回講義の概要) 次回講義で行う演習についての課題の事前説明(講義0.2h)	開発者サイドのヒューマンファクターに影響を与えるツール 開発者のモチベーション	CMP.cf.7 CMP.cf.7	開発者のモチベーション、効率化、コミュニケーションの円滑化に役立つ、グループウェアやビジュアルウェアの役割が説明できる 最新動向(XP、アジャイル、オープンソース)と開発者のモチベーションの関係が把握できる
12	システム提案(1)	具体的なGUIについてシステム設計と評価方法についての計画を立案する		ヒューマンコンピュータインタフェース設計の原則を満たすGUIの設計方針が立案できる
13	システム提案(2)	具体的なGUIについてシステム設計と評価方法について提案を具体化する (Javaで実装してもよいが実装は必須ではない)		ヒューマンコンピュータインタフェース設計の原則を満たすGUIが設計できる
14	重要事項のまとめ			
15	期末試験			
教科書・参考書	岡田、西田、葛岡著:「ヒューマンコンピュータインタラクション」ITTextシリーズ、オーム社 村田著:「ヒューマンインタフェースの基礎と応用」日本出版サービス			
備考				

授業科目名		開発マネジメント					
単位数		2					
開設学期		3年後期					
目的		ソフトウェア開発プロジェクトの概要を理解し、基本的なビジネススキル、マネジメントスキルを学習することを目的とする					
概要		ソフトウェア開発は個人で実施するのではなく複数のプロフェッショナルが協調して、組織的にプロジェクトとして実施する。ソフトウェア開発プロジェクトを円滑に実施するために、各個人が備えるべき基本的なビジネススキルを具体的に理解し、また、プロジェクトマネジメントスキルの概要を学習することを目的とする。					
目標		ソフトウェア開発者が備えるべき基本的なビジネススキル、及び、基本的なプロジェクトマネジメントスキルを理解し、使えることを目標とする					
先修科目							
関連科目							
授業方法							
評価方法・基準							
授業展開							
1	授業の概要と目的および進め方について説明する。(0.3h)ソフトウェア開発プロジェクトの概要を理解し、その中でプロジェクトマネジメントの役割を説明する。また、チーム作業と個人の役割を理解する。	学習成果	トピック	学習目標	発展学習目標		
		MGT.con.1	一般的なプロジェクトマネジメント	MGT.con.1	一般的なプロジェクトマネジメント、ソフトウェア開発は複数人数で行うことを理解する	MGT.con.1	日本の「平均的なプロジェクト」をデータで理解する。(IPAの調査など)
		MGT.con.3	プロジェクトマネジメントの役割	MGT.con.3	プロジェクトマネジメントの役割を説明できる		
		PRF.psy.1	チームやグループでの作業の際のダイナミクス	PRF.psy.1	具体的にチームやグループの作業を理解し、説明できる。		
		PRF.psy.2	個人の認知(制限など)	PRF.psy.2	チームにおける個人の役割を理解し、説明できる。		
2	チームやグループでの作業でのチームと個人の間で認識すべき問題(複雑性、曖昧性など)とそれへの対処を説明する。	PRF.psy.3	認知的問題の複雑性	PRF.psy.3	チームメンバー間での認識の違いなど、複雑性を具体的に理解し説明できる		
		PRF.psy.4	ステークホルダーとの対話	PRF.psy.4	チーム作業に関わるステークホルダーとその対処を理解し、説明できる		
		PRF.psy.5	不確実性と曖昧性の取り扱い	PRF.psy.5	チーム作業での不確実性と曖昧性をどう扱っていくかを理解し、説明できる		
		PRF.psy.6	多国籍・多文化環境の取り扱い	PRF.psy.6	多国籍・多文化環境の具体例とその取り扱いを理解し、説明できる		
3	ソフトウェア開発者のコミュニケーションスキル	PRF.com.1	読解、理解、要約(ソースコードやドキュメントなど)設計という概念の定義	PRF.com.1	ソフトウェア開発に必要な文書を理解し、使用できる		
		PRF.com.2	記述(職務記述書、報告書、評価報告書、理由書など)	PRF.com.2	チームでの作業に必要な書類を理解できる		
4	ソフトウェア開発者のコミュニケーションスキル	PRF.com.3	チームとグループのコミュニケーション(口頭、文書、電子メールなど)複数のソフトウェア開発ライフサイクルにおける設計の関係	PRF.com.3	チームとグループのコミュニケーションに関する複数の手段を理解し、使用できる		
		PRF.com.4	プレゼンテーションスキル	PRF.com.4	プレゼンテーションの具体的なポイントを理解し、使用できる。		
5	ソフトウェア開発者のプロフェッショナリズム、個人の行動規範や資格に関して	PRF.pr.1	アクレディテーション、資格認定、免許制度	PRF.pr.1	ソフトウェア開発に関して推奨されている資格を理解し、説明できる	PRF.pr.1	ITSSに関して理解し、説明できるようにする
		PRF.pr.2	倫理綱領とプロフェッショナルとしての社会的、法的、歴史的およびプロフェッショナルとしての考慮事項	PRF.pr.2	プロフェッショナルの行動に関して理解し、説明できる	PRF.pr.2	事例としてPMIを理解し、説明できるようにする
		PRF.pr.3	プロフェッショナル・ソサエティ(学会や協会、コミュニティなど)の起源と役割	PRF.pr.3	プロフェッショナルの考慮事項に関して理解し、説明できる		
6	ソフトウェア開発者のプロフェッショナリズム、業界の団体やその役割。	PRF.pr.4	標準の起源と役割	PRF.pr.4	プロフェッショナル・ソサエティ(学会や協会、コミュニティなど)の起源と役割を理解し、いくつかの具体的なソサエティを挙げることができる。	PRF.pr.4	国内の情報関係の学会と団体を理解し、説明できるようにする。
		PRF.pr.5	ソフトウェアの経済的重要性	PRF.pr.5	標準の必要性を理解し、具体的な標準を挙げることができる。	PRF.pr.5	ソフトウェア工学に重要な標準(ISO関連等)の事例を理解する
		PRF.pr.6	雇用形態・雇用契約	PRF.pr.6	ソフトウェアの経済的重要性について理解し、説明できる		
		PRF.pr.7		PRF.pr.7	代表的な雇用形態を理解し、説明できる		
7	マネジメントの基礎	MGT.con.1	一般的なプロジェクトマネジメント	MGT.con.1	一般的なプロジェクトマネジメントを理解し、説明できる(再)		
		MGT.con.2	古典的なマネジメントモデル	MGT.con.2	古典的なプロジェクトマネジメントを理解し、説明できる		
		MGT.con.3	プロジェクトマネジメントの役割	MGT.con.3	プロジェクトマネジメントの役割を理解し、その重要性を説明できる(再)		
		MGT.con.4	全社的/組織的なマネジメントの構造	MGT.con.4	全社的/組織的なマネジメントの構造を理解し、代表的な構造を挙げることができる		
		MGT.con.5	ソフトウェア開発マネジメントの種類(調達マネジメント、プロジェクトマネジメント、開発マネジメント、保守マネジメント、リスクマネジメントなど)	MGT.con.5	ソフトウェア開発マネジメントの種類を理解し、各々の概要を簡単に説明できる	MGT.con.5	PMBOKの知識体系の俯瞰図を具体例として理解する。
8	プロジェクト計画の基礎	MGT.pp.1	評価と計画	MGT.pp.1	評価と計画の概要を理解し、説明することができる	MGT.pp.1	PMBOKの「プロジェクトスコープ」を理解する。
		MGT.pp.2	WBS(Work Breakdown Structure)の作成	MGT.pp.2	WBS(Work Breakdown Structure)を理解し、与えられた素材を前提に使用することができる。		
9	タイムマネジメント	MGT.pp.3	タスクのスケジューリング	MGT.pp.3	タスクのスケジューリングを理解し、使用することができる		
		MGT.pp.4	工数の見積もり	MGT.pp.4	工数の見積りの手法を理解し、説明することができる。具体的なコスト見積りの手法(類推見積り、係数モデル見積り、ボトム・アップ見積り)を挙げることができる。		
		MGT.pp.5	リソースの割り当て	MGT.pp.5	リソースの割り当ての手法を理解し、説明することができる。具体的な事例(責任分担マトリクスなど)を使用できるようにする。		
10	リスクマネジメントと品質マネジメント	MGT.pp.6	リスクマネジメント	MGT.pp.6	リスクマネジメントの手法を理解し、説明することができる。リスクの種類とその具体例を説明できる。		
		QUA.pca.2, QUA.pda	品質マネジメント	QUA.pca.2, QUA.pda	プロジェクトマネジメントの立場で品質計画、品質保証、品質管理を理解し、説明することができる。品質管理の手法の具体例(特性要因図、パレート図など)を説明できる。		

授業展開	学習成果	トピック	学習目標	発展学習目標	
11	プロジェクトメンバと組織	MGT.per.1	組織構造, 地位, 責任, 権限	MGT.per.1 組織構造, 地位, 責任, 権限を理解し, 代表的な組織の構造(プロジェクト型, 機能型, マトリクス型)を挙げて, その性質を説明することができる。	
		MGT.per.2	フォーマルおよびインフォーマルなコミュ	MGT.per.2 フォーマルおよびインフォーマルなコミュニケーションを理解し, 説明することができる	
		MGT.per.3	メンバの割り当て	MGT.per.3 メンバの割り当てを具体的に理解し, 使用することができる。メンバに対する要求事項があることを理解する。	
		MGT.per.4	メンバの教育, キャリア開発, 評価	MGT.per.4 メンバの教育, キャリア開発, 評価を理解し, 説明することができる	
		MGT.per.5	会議のマネジメント	MGT.per.5 会議のマネジメントの技術を理解し, 説明することができる	
		MGT.per.6	チームビルディングとモチベーション向上	MGT.per.6 チームビルディングとモチベーション向上にむけた技術を理解し, 使用することができる	
		MGT.per.7	対立の解消	MGT.per.7 対立の解消のための技術を理解し, 使用することができる	
12	プロジェクトのコントロール	MGT.ctl.1	変更のコントロール	MGT.ctl.1 変更のコントロールに関して理解し, 説明することができる。特にスコープ変更管理について具体的に説明できるようにする。	
		MGT.ctl.2	モニタリングと報告	MGT.ctl.2 モニタリングと報告に関して理解し, 具体的な技術を挙げる事ができる	
		MGT.ctl.3	結果の測定と分析	MGT.ctl.3 結果の測定と分析に関して理解し, 具体的な技術を挙げる事ができる #使用したい	
		MGT.ctl.4	是正と挽回	MGT.ctl.4 是正と挽回に関して理解し, 具体的な技術とその効果を挙げる事ができる	
		MGT.ctl.5	インセンティブと規律	MGT.ctl.5 インセンティブと規律の重要性を理解し, 説明することができる	
		MGT.ctl.6	実績の基準	MGT.ctl.6 実績の基準を理解し, 具体的にいくつかを挙げる事ができる	
13	ソフトウェア構成管理	MGT.cm.1	リビジョン管理	MGT.cm.1 リビジョン管理を理解し, 説明することができる	
		MGT.cm.2	リリース管理	MGT.cm.2 リリース管理を理解し, 説明することができる	
		MGT.cm.3	支援ツール	MGT.cm.3 支援ツールの重要性を理解し, 代表的なツールを挙げる事ができる	
		MGT.cm.4	ビルド	MGT.cm.4 ビルドを理解し, 代表的なツールを挙げる事ができる	
		MGT.cm.5	構成管理のプロセス	MGT.cm.5 構成管理のプロセスを理解し, 代表的なプロセスを説明することをできる	
		MGT.cm.6	保守に関する考慮事項	MGT.cm.6 保守に関する考慮事項を理解し, 説明することができる	
		MGT.cm.7	分散環境とバックアップ	MGT.cm.7 optional 分散環境でのバックアップに関して理解し, 説明することができる	
14	重要事項のまとめ				
15	期末試験				
教科書・参考書	参考書「プロジェクトマネジメント知識体系ガイド 第3版 (PMBOK®ガイド第3版)」 発行:PMI				
備考					



授業科目名	プログラミング基礎実習
単位数	2
開設学期	2年後期
目的	工学的なソフトウェア開発の一環としてプログラムを構築する技術を身につけることを目的とする。
概要	講師が提示する例題ソフトウェアに対する概略設計内容を入力として、プログラム開発と単体テスト、統合テストを実施する。また、開発したプログラムや実施したテストに関する仕様の文書化も行う。実習にあたり、(主にプログラミングのための)開発支援ツールなどを活用することで、工学的なプログラム開発を体験する。 (※注)授業展開や各項目の配分は固定的なものではなく、例題の性質、想定するプログラミングパラダイムなどに応じて柔軟に編成されるべきである。
目標	ソフトウェアの設計内容が与えられたときに、開発支援ツールなどを活用しながら、主に個人でモジュール分割を行い、それに従ってプログラムを開発し、開発したプログラムをテストする能力を身につける。また、自身のプログラミング作業を管理し、開発したプログラムの仕様書やテスト仕様書を記述する能力を身につける。
先修科目	プログラミング入門、プログラミング基礎、ソフトウェア構築
関連科目	ソフトウェア設計
授業方法	演習
評価方法・基準	下記により評価する。 ・実習への取り組み姿勢と発表成果 ・実習成果物

授業展開	学習成果	トピック	学習目標
1	実習の概要と目的および進め方について説明する。	実習の全体計画	実習で行うプログラミングプロセスの全体像を理解する。
	実習課題について説明する。	開発対象の理解	開発対象の設計仕様の概要を理解する。
	実習に用いるツールなどの使用法を理解する。	開発ツールの理解	ごく小規模なプログラムを実際に開発するなどし、開発に使用するツールなどを試用し、使用法の概略を理解する。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
2	設計のレビューを行う。	設計のレビュー	課題として与えられた設計仕様書をレビューし、不明な点や疑問点を解消する。レビュー報告書を作成する。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
3	モジュール分割を実施する。	モジュール分割の実施	特定のパラダイム(例、オブジェクト指向、構造化)に従い、モジュール分割を実施し、それぞれに対するモジュール仕様書を作成する。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
4	モジュール仕様書をレビューする。	モジュール仕様書のレビュー	前回行ったモジュール分割結果をレビューし、必要に応じてモジュール分割結果やモジュール仕様書の修正を行う。レビュー報告書を作成する。
	単体テストを設計する。	単体テストの設計	モジュール分割結果からそれぞれのモジュールに対する単体テストを設計し、単体テスト仕様書を作成する。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
5	作業計画を立案する。	作業計画の立案	プログラミングと単体テスト、モジュール統合および統合テストの計画を立案し、レビューする。
	プログラミングと単体テストを実施する(1)	プログラミングと単体テストの実施	立案した計画に従い、プログラミングと単体テストを実施する。プログラミング結果をプログラム仕様書、単体テスト結果を単体テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況进行评估する。その結果に基づいて作業報告書を作成する。
6	プログラミングと単体テストを実施する(2)	プログラミングと単体テストの実施	立案した計画に従い、プログラミングと単体テストを実施する。プログラミング結果をプログラム仕様書、単体テスト結果を単体テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況进行评估する。その結果に基づいて作業報告書を作成する。
7	プログラミングと単体テストを実施する(3)	プログラミングと単体テストの実施	立案した計画に従い、プログラミングと単体テストを実施する。プログラミング結果をプログラム仕様書、単体テスト結果を単体テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況进行评估する。その結果に基づいて作業報告書を作成する。
8	プログラミングと単体テストを実施する(4)	プログラミングと単体テストの実施	立案した計画に従い、プログラミングと単体テストを実施する。プログラミング結果をプログラム仕様書、単体テスト結果を単体テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況进行评估する。その結果に基づいて作業報告書を作成する。

授業展開	学習成果	トピック	学習目標
9	モジュールの統合と統合テストを実施する(1)	モジュール統合と統合テストの実施	立案した計画に従い、統合テストを実施する。統合テスト結果を統合テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況を評価する。その結果に基づいて作業報告書を作成する。
10	モジュールの統合と統合テストを実施する(2)	モジュール統合と統合テストの実施	立案した計画に従い、統合テストを実施する。統合テスト結果を統合テスト報告書としてまとめる。
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業の進捗状況を評価する。その結果に基づいて作業報告書を作成する。
11	プログラムの評価を行う(1)	プログラム評価の実施	作成したプログラムに対して、一定の基準に従った評価を行う。たとえば、静的解析や動的解析、性能解析、メトリクス計測を行い、結果を報告書としてまとめる。、
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
12	プログラムの評価を行う(2)	プログラム評価の実施	作成したプログラムに対して、一定の基準に従った評価を行う。たとえば、静的解析や動的解析、性能解析、メトリクス計測を行い、結果を報告書としてまとめる。、
	実習で行った作業を振り返る。	作業の振り返り	今回行った作業を振り返り、作業報告書を作成する。
13	成果発表会(1)	演習成果の発表	作成したプログラムとテスト結果、評価・解析結果を報告する。
14	成果発表会(2)		
15	全体のまとめ		
教科書・参 考書 備			

授業科目名	プログラミング応用実習
単位数	2
開設学期	3年前期
目的	工学的なソフトウェア開発の一環としてプログラムを構築する技術を身につけることを目的とする。応用実習では、例題として与えられるドメイン知識の理解に基づいてプログラムを開発する技術や、既存のソフトウェア資産を再利用する技術を習得する。
概要	講師が提示する例題ソフトウェアシステムに対する要求仕様書を入力として、ソフトウェアアーキテクチャの設計とプログラム開発、テストを実施する。また、実習にあたり、統合開発環境などを活用することで、工学的なプログラム開発を体験する。 (※注)授業展開や各項目の配分は固定的なものではなく、例題のドメインが持つ性質、再利用資産の利用形態などに応じて柔軟に編成されるべきである。
目標	ソフトウェアの要求仕様が与えられたときに、統合開発環境やプロセス管理ツール、構成管理ツールなどを活用しながら、グループまたは個人でソフトウェアアーキテクチャの設計やプログラム開発、テスト、保守する能力を身につける。また、自身またはグループによる開発作業を管理するとともに開発したソフトウェア成果物の管理をする能力を身につける。
先修科目	プログラミング基礎演習、ソフトウェア設計
関連科目	モデル化と要求開発、ソフトウェアアーキテクチャ、開発マネジメント
授業方法	演習
評価方法・基準	下記により評価する。 ・実習への取り組み姿勢と発表成果 ・実習成果物

授業展開	学習成果	トピック	学習目標
1	実習の概要と目的および進め方について説明する。 実習課題について説明する。 実習に用いるツールなどの使用法を理解する。 実習で行った作業を振り返る。	実習の全体計画 開発対象の理解 開発環境・使用ツールの理解 作業の振り返り	実習で行う開発プロセスの全体像を理解する。 開発対象の要求仕様の概要を理解する。グループによる開発を行う場合には開発体制(メンバと役割)を決定する。 ごく小規模なプログラムを実際に開発するなどし、開発に使用する環境やツールを試用し、使用法の概略を理解する。 今回行った作業を振り返り、作業報告書を作成する。
2	設計のレビューを行う。 実習で行った作業を振り返る。	設計のレビュー 作業の振り返り	課題として与えられた要求仕様書をレビューし、不明な点や疑問点を解消する。レビュー報告書を作成する。また、必要なドメイン知識や再利用資産に関して調査し、必要に応じて要求仕様書を補足する文書を作成する。 今回行った作業を振り返り、作業報告書を作成する。
3	作業計画を立案する。 ソフトウェアアーキテクチャを設計する(1) 実習で行った作業を振り返る。	作業計画の立案 サブシステムへの分割、サブシステム間の通信方式の決定 作業の振り返り	開発作業計画を立案する。グループ開発の場合は役割分担を行う。 ソフトウェアシステムをサブシステムに分割し、それぞれの間の通信方式を決定する。ソフトウェアアーキテクチャの文書化を行い概略仕様書としてまとめる。またシステムテスト仕様書の作成を行う。ドメイン知識や再利用資産を考慮し、アーキテクチャを評価し、必要に応じて再設計を行う。 今回行った作業を振り返り、作業の進捗状況进行评估する。必要に応じて再計画を行う。その結果に基づいて作業報告書を作成する。
4	実習で行った作業を振り返る。	サブシステムへの分割、サブシステム間の通信方式の決定 作業の振り返り	ソフトウェアシステムをサブシステムに分割し、それぞれの間の通信方式を決定する。ソフトウェアアーキテクチャの文書化を行い概略仕様書としてまとめる。またシステムテスト仕様書の作成を行う。ドメイン知識や再利用資産を考慮し、アーキテクチャを評価し、必要に応じて再設計を行う。 今回行った作業を振り返り、作業の進捗状況进行评估する。必要に応じて再計画を行う。その結果に基づいて作業報告書を作成する。
5	ソフトウェアアーキテクチャを設計する(3) 実習で行った作業を振り返る。	サブシステムへの分割、サブシステム間の通信方式の決定 作業の振り返り	ソフトウェアシステムをサブシステムに分割し、それぞれの間の通信方式を決定する。ソフトウェアアーキテクチャの文書化を行い概略仕様書としてまとめる。またシステムテスト仕様書の作成を行う。ドメイン知識や再利用資産を考慮し、アーキテクチャを評価し、必要に応じて再設計を行う。 今回行った作業を振り返り、作業の進捗状況进行评估する。必要に応じて再計画を行う。その結果に基づいて作業報告書を作成する。
6	プログラムの設計と構築を行う(1) 実習で行った作業を振り返る。	モジュールへの分割とプログラミング、単体テスト、統合テストの実施 作業の振り返り	各サブシステムの詳細設計、モジュール分割、プログラミング、単体テスト、統合テストを行い、それぞれを仕様書、報告書としてまとめる。必要に応じてドメイン知識や再利用資産に合わせた設計変更を行う。 今回行った作業を振り返り、作業の進捗状況进行评估する。必要に応じて再計画を行う。その結果に基づいて作業報告書を作成する。

授業展開	学習成果	トピック	学習目標
7	プログラムの設計と構築を行う(2)	モジュールへの分割とプログラミング, 単体テスト, 統合テストの実施	各サブシステムの詳細設計, モジュール分割, プログラミング, 単体テスト, 統合テストを行い, それぞれを仕様書, 報告書としてまとめる. 必要に応じてドメイン知識や再利用資産に合わせた設計変更を行う.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業の進捗状況を評価する. 必要に応じて再計画を行う. その結果に基づいて作業報告書を作成する.
8	プログラムの設計と構築を行う(3)	モジュールへの分割とプログラミング, 単体テスト, 統合テストの実施	各サブシステムの詳細設計, モジュール分割, プログラミング, 単体テスト, 統合テストを行い, それぞれを仕様書, 報告書としてまとめる. 必要に応じてドメイン知識や再利用資産に合わせた設計変更を行う.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業の進捗状況を評価する. 必要に応じて再計画を行う. その結果に基づいて作業報告書を作成する.
9	プログラムの設計と構築を行う(4)	モジュールへの分割とプログラミング, 単体テスト, 統合テストの実施	各サブシステムの詳細設計, モジュール分割, プログラミング, 単体テスト, 統合テストを行い, それぞれを仕様書, 報告書としてまとめる. 必要に応じてドメイン知識や再利用資産に合わせた設計変更を行う.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業の進捗状況を評価する. 必要に応じて再計画を行う. その結果に基づいて作業報告書を作成する.
10	システムテストを実施する(1)	システムテスト(主に非機能要求のテスト)を行う	システムテスト計画に従い, システムテストを実施する. 結果を報告書としてまとめる.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業の進捗状況を評価する. 必要に応じて再計画を行う. その結果に基づいて作業報告書を作成する.
11	システムテストを実施する(2)	システムテスト(主に非機能要求のテスト)を行う	システムテスト計画に従い, システムテストを実施する. 結果を報告書としてまとめる.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業の進捗状況を評価する. 必要に応じて再計画を行う. その結果に基づいて作業報告書を作成する.
12	ソフトウェアシステムの評価を行う	システム評価の実施	作成したプログラムに対して, 一定の基準に従った評価を行う. たとえば, 静的解析や動的解析, 性能解析, メトリクス計測を行い, 結果を報告書としてまとめる.
	開発実績の評価を行う	開発実績の評価	これまでに行ってきた開発の実績を振り返る. 当初の計画とのずれや, 開発途中での計画変更要因について考察し, 報告書としてまとめる.
	実習で行った作業を振り返る.	作業の振り返り	今回行った作業を振り返り, 作業報告書を作成する.
13	成果発表会(1)	演習成果の発表	作成したプログラムとテスト結果, 評価・解析結果を報告する.
14	成果発表会(2)		
15	全体のまとめ		
教科書・参考書			
備考			

授業科目名	エンタープライズソフトウェア開発実習		
単位数	2 (ただし実習科目のため45時間)		
開設学期	3年後期		
目的	これまでに学んだ要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストの技法を用いて、永続化データの処理を中心とする実際のエンタープライズソフトウェアシステムを対象として開発工程を通して行い、各工程の作業内容、及び工程間の関係を理解することを目標とする		
概要	講師が提示する例題システムに対する要望を入力として、要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストを実施する。グループでの演習を効率良く実施するために実習冒頭でグループのリーダー役を定め、リーダーが中心となって、作業計画と作業目標を実習単位で明確にし、各実習終了前に計画の進捗度合いを確認し、計画に遅延が生じている場合は計画の再調整を実施する。演習にあたりグループの構成員全員が貢献することが望ましい。		
目標	<ul style="list-style-type: none"> <li>・エンタープライズソフトウェアシステム開発における各工程の作業内容が理解できる</li> <li>・エンタープライズソフトウェアシステム開発工程のV字の関係が理解できる</li> <li>・グループでの作業を効率良く進めるためのプロジェクトマネジメントにおける計画マネジメント、コミュニケーションマネジメントを体感する。</li> </ul>		
先修科目			
関連科目			
授業方法	グループ演習		
評価方法・基準	下記等により評価する。 ・グループ演習への取り組み姿勢とグループ発表成果 ・グループ演習成果物 ・個人単位で提出される成果レポートの内容		
授業展開	トピック	学習目標	
1	演習の概要と目的および進め方について説明し、実習のコマ数と各コマの作業目標(想定)を発表する。グループ演習用のグループ分けも発表する。システム要求分析工程およびシステムアーキテクチャ設計工程として、ソフトウェアシステム化の背景と要望を理解し、作業計画と目標を立案する。	演習の全体計画	実習の進め方と各コマの達成目標(案)、各工程で作成すべき成果物を理解する、グループのリーダーを選定する
	ソフトウェアシステム化の背景と要望		システム化の背景と想定顧客の要望を理解する
	作業計画と目標の立案		各コマ単位の作業計画と目標を立案する
	開発ツールの使用説明		使用するモデリングツールの操作方法を理解する
	ソフトウェア要求の抽出(1)		要求の抽出を行うにあたり、曖昧な点等の聞き出したい項目を列挙し、顧客役に伝える(教員等が顧客役を担当する)。
2	ソフトウェア要求分析工程の作業成果物を作成する(1)	ソフトウェア要求の抽出(2)	要求の抽出を行うにあたり、事前に伝えた項目に基づいて顧客役から曖昧な点等を聞き出す。顧客役は事前に伝えられた項目を整理して返答を準備しておくことが望ましい。
		ソフトウェア要求の分析	抽出した要求を分析し、システム化前の業務フロー図とシステム化後の業務フロー図を作成する
			抽出した要求から画面レイアウト、画面遷移図を作成する
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する

授業展開		トピック	学習目標
3	ソフトウェア要求分析工程の作業成果物を作成する(2)	要求の仕様化	システム化後の業務フロー図をベースに要求をモデル化する(例えばユースケース図およびユースケース記述)。 要求のモデルに基づいて、システム化対象を選定する(各グループ1機能の実装を前提とする) 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
4	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(1)	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換(1)	選定した機能について、システム化の方法を検討し、必要な静的構造をモデル化する(例えば、エンティティクラスのクラス図、および、ロバストネス分析の実施によるロバストネス図の作成)。 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
5	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(2)	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換(2)	選定した機能について、必要な動的振る舞いをモデル化する(例えば、分析シーケンス図や分析クラス図の作成)。 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
6	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(1)	アーキテクチャ設計工程における最上位設計(1)	選定した機能について、必要な静的構造および動的振る舞いを設計し、その結果をモデル化する(例えば設計シーケンス図や設計クラス図の作成)。 グループ計画に対する進捗を確認し、計画変更の必要性を検討する
7	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(2)	アーキテクチャ設計工程における最上位設計(2)	非機能要求や制約・実現環境を考慮して、アーキテクチャ設計を行う その他の設計書(画面定義書、処理定義書、SQL定義書など)を作成する 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
8	データベース設計を実施する(1)	データベース設計(1)	トップダウン分析による概念データモデル設計を実施する ボトムアップ分析による概念データモデル設計を実施する 両者を統合し、正規化し、概念データモデルを完成させる 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
9	データベース設計を実施する(2)	データベース設計(2)	概念データモデルをベースに論理データモデルを設計する 成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する

授業展開		トピック	学習目標	
10	ソースコード作成及びテスト工程の作業成果物を作成する	ソースコード作成およびテスト		配布するソースコードとコーディング規約を用いてソースコードレビューを実施する
				単体テスト計画書を作成し、テスト項目を抽出する
				成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
11	テスト工程の作業成果物を作成する(1)	ソフトウェア単体テスト		配布するソースコードに対して単体テストを実施する
				単体テスト故障処理票を作成する
				単体テスト報告書を作成する
				成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
12	テスト工程の作業成果物を作成する(2)	ソフトウェア結合テストおよびシステム統合テスト		ソフトウェア結合テスト計画書およびシステム結合テスト計画書を作成し、テスト項目を抽出する
				配布するソースコードに対してソフトウェア結合テストおよびシステム結合テストを実施する
				ソフトウェア結合テスト故障処理票およびシステム結合テスト故障処理票を作成する
				ソフトウェア結合テスト報告書およびシステム結合テスト報告書を作成する
				成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
13	成果発表会(1)			グループ単位で発表の準備を実施する
				成果として得られたシステムを利用する様子を実演(デモンストレーション)し、顧客役が実演を確認することをもってシステム適格性確認テスト(検収)を受ける。
				授業開始時に立案した作業計画と目標に対する実績を発表し、計画との相違が生じた場合はその原因を分析し改善策を示す
				各工程の作業成果物を提示する
14	成果発表会(2)			グループごとに成果を発表する
15	成果発表会(3)および全体のまとめ			グループごとに成果を発表し、最後に全体の内容をまとめる。
教科書・参考書	「実践で学ぶソフトウェア開発」 (株)NTTデータソフトウェア工学推進センタ著 オーム社(2008.2)			
備考				

授業科目名	組込みソフトウェア開発実習		
単位数	2(ただし実習科目のため45時間)		
開設学期	3年後期		
目的	これまでに学んだ要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストの技法を用いて実際の組込みソフトウェアシステム開発工程を通して行い、組込み開発特有の制約や性質を加味した各工程の作業内容、及び工程間の関係を理解する。		
概要	講師が提示する例題システムに対する顧客の要望を入力として、組込みソフトウェアシステム開発における要求分析、アーキテクチャ設計、ソースコード作成/プログラミング、テストを実施する。グループでの演習を効率良く実施するために講義冒頭でグループのリーダー役を定め、リーダーが中心となって、作業計画と作業目標を講義単位で明確にし、各講義終了前に計画の進捗度合いを確認し、計画に遅延が生じている場合は計画の再調整を実施する。演習にあたりグループの構成員全員が貢献することが望ましい。		
目標	<ul style="list-style-type: none"> <li>・組込みソフトウェアシステム開発における各工程の作業内容が理解できる。</li> <li>・組込みソフトウェアシステム開発工程のV字の関係が理解できる。</li> <li>・グループでの作業を効率良く進めるためのプロジェクトマネジメントにおける計画マネジメント、コミュニケーションマネジメントを体感する。</li> <li>・リアルタイム性や省メモリ、クロス開発といった組込み開発特有の性質および制約を理解し、それらを考慮した開発技法を習得する。</li> </ul>		
先修科目	ハードウェア制御・リアルタイムシステム(組込み・リアルタイムシステム)(SAS.emb)		
関連科目			
授業方法	グループ演習		
評価方法・基準	下記等により評価する。 ・グループ演習への取組み姿勢とグループ発表成果 ・グループ演習成果物 ・個人単位で提出される成果レポートの内容		
授業展開	トピック	学習目標	
1	演習の概要と目的および進め方について説明し、実習のコマ数と各コマの作業目標(想定)を発表する。システム要求分析工程およびシステムアーキテクチャ設計工程として、システム化の背景と要望を理解し、作業計画と目標を立案する。	演習の全体計画	実習の進め方と各コマの達成目標(案)、各工程で作成すべき成果物を理解する、グループのリーダーを選定する
	システム化の背景と要望および要求		システム化の背景、利用可能なハードウェア環境、外界・自然物理現象、想定顧客の要望を理解する
	作業計画と目標の立案		各コマ単位の作業計画と目標を立案する
	モデリングツールの理解		使用するモデリングツールの操作方法を理解する
	システム要求の分析		システム化にあたり理解した事柄を整理して、組込みシステム全体に対する機能要求と非機能要求を抽出しモデル化する(例えばユースケース図、ユースケース記述、コンテキスト図の利用)。
2	システム要求分析工程およびシステムアーキテクチャ設計工程の作業成果物を作成する	システム設計・レビュー	システムの要求モデルに基づいてハードウェアとソフトウェアの責務分担を検討し得られるシステムアーキテクチャの設計結果を配布し、要求に照らしてレビューし、システム要求に対して構造および振る舞いが検討・モデル化(例えばブロック図、ロバストネス図、クラス図、ステートチャート図によって記述)されていることを理解する。
		ハードウェア設計・レビュー	システムアーキテクチャから導出されるハードウェア設計に従う参照ハードウェアを配布し、幾つかの初歩的なハードウェア制御実験を通じてその特性を理解し、当該ハードウェアがシステム要求およびアーキテクチャに基づいて適切に選択・設計・実現されていることを理解する。
		ソフトウェア要求の抽出(1)	システムへと組み込むソフトウェアに対する要求の抽出を行うにあたり、曖昧な点等の聞き出したい項目を列挙し、顧客役に伝える(教員等が顧客役を担当する)。



授業展開	トピック	学習目標	
3	ソフトウェア要求分析工程の作業成果物を作成する(1)	ソフトウェア要求の抽出(2)	ソフトウェア要求の抽出を行うにあたり、事前に伝えた項目に基づいて顧客役から曖昧な点等を聞き出す。顧客役は事前に伝えられた項目を整理して返答を準備しておくことが望ましい。
		ソフトウェアの機能要求の分析と仕様化	抽出した要求を分析し、機能要求およびハードウェアや外部環境との関係・相互作用をモデル化する(例えばUMLユースケース図、ユースケース記述、イベントリスト、コンテキスト図の利用)。
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
4	ソフトウェア要求分析工程の作業成果物を作成する(2)	ソフトウェアの非機能要求の分析と仕様化	ソフトウェアに対する異常・例外系を含む品質要求や制約をモデル化する(例えば非機能要求リスト、タイミング仕様書、UMLタイミング図の利用)。
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
5	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(1)	ソフトウェアアーキテクチャ設計工程におけるソフトウェア要求からソフトウェアアーキテクチャへの変換(1)	選定した機能について、ソフトウェア化の方法を検討し、必要な静的構造をモデル化する(例えば、エンティティクラスのクラス図、および、ロバストネス分析の実施によるロバストネス図の作成)。
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
6	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換タスクの作業成果物を作成する(2)	ソフトウェアアーキテクチャ設計工程における要求からアーキテクチャへの変換(2)	選定した機能について、必要な動的振る舞いをモデル化する(例えば、分析シーケンス図や分析クラス図、状態遷移表、分析状態チャート図の作成)。
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
7	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(1)	アーキテクチャ設計工程における最上位設計(1)	選定した機能について、必要な静的構造および動的振る舞いを設計し、その結果をモデル化する(例えば設計シーケンス図や設計クラス図の作成)。
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
8	アーキテクチャ設計工程における最上位設計タスクの作業成果物を作成する(2)	アーキテクチャ設計工程における最上位設計(2)	非機能要求を考慮して、アーキテクチャ設計を行う
			その他の設計書(タスク構造設計、メモリ構造設計など)を作成する
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
9	ソースコード作成及びテスト工程の作業成果物を作成する	ソースコード作成およびテスト	配布するソースコードに対して、コーディング規約およびシステム・プログラミング環境上の特性(例えば省メモリ)を考慮してソースコードレビューを実施する
		クロス開発環境の理解	使用するクロス開発環境の特性と操作方法を理解する
			単体テスト計画書を作成し、テスト項目を抽出する
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
10	テスト工程の作業成果物を作成する(1)	ソフトウェア単体テスト	配布するソースコードに対して単体テストを実施する
			単体テスト故障処理票を作成する
			単体テスト報告書を作成する
			成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する

授業展開	トピック	学習目標
11	テスト工程の作業成果物を作成する (2)	ソフトウェア結合テスト ソフトウェア結合テスト計画書を作成し、テスト項目を抽出する ソフトウェア結合テストを実施し、故障処理票およびテスト報告書を作成する
	システム統合テスト	システム結合テスト計画書を作成し、テスト項目を抽出する ソースコードをコンパイルして配布ハードウェア上に配備し、システム結合テストを実施し、故障処理票およびテスト報告書を作成する
		成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
12	保守工程の作業成果物を作成する	システム適格性確認テスト 配布ハードウェアを含むシステム全体の稼働させて制御の正しさや性能、機能の妥当性を検証・測定し、システム要求に照らして適格性を確認し、結果をテスト報告書として作成する。 システム要求に照らして重大な問題が生じた場合は、その原因を追究してソフトウェア設計・実装の修正により対処可能であるかを検討し、可能な場合は対策（例えばチューニング）を施し、再度のシステム適格性確認テストにより改善されたことを確認する。
	システムの問題把握および修正分析、保守	成果物をレビューし、グループ計画に対する進捗を確認し、計画変更の必要性を検討する
13	成果発表会 (1)	グループ単位で発表の準備を実施する 成果として得られたシステムを利用する様子を実演(デモンストレーション)し、顧客役が実演を確認することをもってシステム適格性確認テスト(検収)を受ける。 授業開始時に立案した作業計画と目標に対する実績を発表し、計画との相違が生じた場合はその原因を分析し改善策を示す
		各工程の作業成果物を提示する
14	成果発表会(2)	グループごとに成果を発表する
15	成果発表会(3)および全体のまとめ	グループごとに成果を発表し、最後に全体の内容をまとめる。
教科書・参考書	「実践で学ぶソフトウェア開発」 (株)NTTデータソフトウェア工学推進センター著、オーム社(2008.2) 「組込みソフトウェア開発のためのオブジェクト指向モデリング」 SESSAME WG2著、翔泳社(2006.6) 「組込みソフトウェア開発のための構造化モデリング」 SESSAME WG2著、翔泳社(2006.1) 「組込みソフトウェア開発基礎講座」 杉浦英樹、橋本隆成 著、翔泳社(2007.1)	
備考		

授業科目名	インターンシップ	
単位数	2(ただし演習科目のため、2コマ分、全45時間)	
開設学期	3年生夏季休暇	
目的	民間企業等との連携により、現実のソフトウェア開発プロジェクトを教材とした実践的な学習を行うことを目的とする。	
概要	民間企業等との連携により、現実のソフトウェア開発プロジェクトを教材とした実践的な学習を行う。単なる例題にとどまらない現実のソフトウェア開発作業に関わることにより、納期、予算、組織といった実社会の制約を踏まえたソフトウェア開発の実際について学ぶ。	
目標	開発するシステムの具体化に関わり、ソフトウェアの役割を明確化することができる。企業等の担当者と協調して開発作業に関わり、実開発における諸制約を理解できる。成果について発表、討論ができる。	
先修科目		
関連科目		
授業方法		
評価方法・基準	下記により評価する。 ・事前報告書 ・事後報告書 ・発表、討論の内容 ・インターンシップ受け入れ先の評価	
授業展開		学習成果
	テーマを選択する際に、インターンシップの目的、担当したい作業とその理由、身につけたい能力とその理由などについて、事前に報告書を作成する。全期間を通じ、各自選択したテーマにおけるソフトウェア開発を連携企業等とともに進める。企業等の担当者、指導教員との議論を通じ、ソフトウェア開発において必要となる思考法、問題解決法について総合的に学ぶ。実社会の制約を踏まえた開発作業を通じて得られた成果および、事前報告書に記載した事項が身についたか、などを事後報告書としてまとめ、発表討論を行う。	
教科書・参考書		

授業科目名	<b>卒業研究</b>
単位数	未指定(大学により定めること。学校教育法・大学設置基準。)
開設学期	4年生前期・後期
目的	ソフトウェア工学またはソフトウェア工学に関連する特定の課題について研究または制作し、研究の推進方法や知識および独創的な考えの展開能力を習得することを目的とする。
概要	在学中に修得したソフトウェア工学に関する知識や技術のすべてを、初めての新しい課題に集中して独自の解を求め、その研究成果を論文または制作物の形にまとめ、発表する。これを通して、研究の推進方法、関連研究の調査による知識の進歩、および、新しい独創的な考えを展開してゆく能力を養成する。
目標	ソフトウェア工学に関する研究の推進方法、関連研究の調査による知識、および、新しい独創的な考えを展開してゆく能力を習得すること。
先修科目	
関連科目	
授業方法	
評価方法・基準	
授業展開	全期間を通じ、各自選択したテーマにおける研究または制作を指導教員の指導のもとに進める。指導教員等との議論を通じ、特に先進的な研究または制作課題の達成において必要となる創造的思考法、問題解決法について総合的に学ぶ。研究または制作作業を通じて得られた成果を論文または制作物としてまとめ、発表討論を行う。
教科書・参考書	
備考	