

大学学部等における ソフトウェアエンジニアリング 教育・訓練カリキュラムモデル例「Jpn1」

2004/2/6

(2004/12/8 更新)

(社) 情報処理学会

アクレディテーション委員会

ソフトウェアエンジニアリング・アクレディ
テーション分科会

1. はじめに

この文書は、情報処理学会が委員として参加している、IEEE/ACM Computing Curricula/Software Engineering Steering Committee(CCSE)の活動のなかで、日本におけるソフトウェアエンジニアリング教育・訓練カリキュラムモデルの一例「Jpn1」として作成されたものを、日本語によって紹介したものである。Jpn1はCCSEのSE2004 Volumeから掲載されている。

CCSEの活動は、CCSE: Computing Curricula Software Engineeringで紹介されているとおり、ABETによるアクレディテーションとは独立に行なわれてきた。したがって、Jpn1は、国内外のアクレディテーション審査に直接関連をもつものではない。大学などにおいてソフトウェアエンジニアリング関連の学科やコースなどを設置する際の参照モデルになることを願って作成されたものである。

2. 本カリキュラムを構成する科目

本カリキュラムは、大きく分けて5種類の科目群から構成される。一般教養科目、情報科学基礎科目、ソフトウェアエンジニアリング科目、SAS(System and Application Specialties)科目、演習・卒業研究である。

本カリキュラムの総単位数は128単位、総時間数は1800時間である。以下本章では、本カリキュラムを構成する科目の概要について述べる。詳細は附表(IPSJ-SE推奨カリキュラム表)を参照のこと。

2.1 一般教養科目

本カリキュラムでは、一般教養科目に 50 単位 562.5 時間を割り当てている。一般教養科目の内容は、数学、自然科学、人文社会科学、情報技術など各大学や学部、学科、コースなどでそれぞれ定めることとし、本カリキュラムでは定めない。ただし JABEE における学習保証時間に関する基準[1]及び ABET/EAC における Professional Components[2]との関係から、人文科学、社会科学等（語学教育を含む）を 23 単位 258.75 時間、数学、自然科学を 27 単位 303.75 時間割り当てる必要がある。

2.2 情報科学基礎科目

本カリキュラムでは、情報科学基礎科目に 19 科目 37 単位 517.5 時間を割り当てている。情報科学基礎科目の内容を表 1 に示す。

表 1 情報科学基礎科目

科目名	単位数
コンピュータ入門	2.0
プログラミング及び演習(1)	1.5
プログラミング及び演習(2)	1.5
離散数学と代数構造	2.0
論理と推論	2.0
確率・統計	2.0
情報理論及び符号理論	2.0
形式言語と計算理論	2.0
データ構造とアルゴリズム	2.0
プログラミング言語	2.0
電子回路及び論理回路	2.0
コンピュータアーキテクチャ	2.0
オペレーティングシステム及び演習	2.0
デジタル通信とコンピュータネットワーク及び演習	2.0
コンパイラ及び演習	2.0
データベース基礎及び演習	2.0
HCI とヒューマンファクター	2.0
信頼性工学	2.0
システム工学と問題解決法	2.0

2.3 ソフトウェアエンジニアリング科目

本カリキュラムでは、ソフトウェアエンジニアリング科目に 13 科目 26 単位 371.25 時間を割り当てている。ソフトウェアエンジニアリング科目の内容を表 2 に示す。

表 2 ソフトウェアエンジニアリング科目

科目名	単位数	科目説明
SE1 構築技術とツール及び演習	2.0	構築技術とツールについて学ぶ。構築技術やツール、設計の基本概念、コンポーネントの設計などを扱う。またツールを用いた演習を行う。
SE2 形式手法とモデル論	2.0	形式手法とモデル論について学ぶ。仕様記述言語、正当性検証、モデリングの概念、事前条件や事後条件などを扱う。
SE3 工学基礎	2.0	工学基礎について学ぶ。実験的手法、統計解析、能力評価、セキュリティ、スケーラビリティ、工学的設計の基本概念、ソフトウェア工学以外の分野における工学などを扱う。
SE4 エンジニアリングエコノミクス	2.0	エンジニアリングエコノミクスについて学ぶ。ライフサイクルにおける価値の変化、参加型デザインやステークホルダー分析、QFD などによる製品企画、費用対効果分析、価値評価、経済的な影響などを扱う。
SE5 組織論とマネジメント	2.0	組織論とマネジメントについて学ぶ。グループダイナミクス、人間の心理、マネジメントの基本概念、プロジェクトマネジメント、構成管理などを扱う。
SE6 コミュニケーションスキル及び演習	2.0	コミュニケーションと要求獲得の技術について学ぶ。コミュニケーションスキル、インタビュー、ブレインストーミング、プロトタイピング、仕様化とドキュメンテーション、表現、折衝などを扱う。また演習を行う。
SE7 ソフトウェアモデル及び演習	2.0	ソフトウェアモデリングについて学ぶ。モデリングの基本原則、さまざまなモデル、要求の評価、要求の管理などを扱う。
SE8 ソフトウェア設計 (1) 及び演習	2.0	ソフトウェア設計について学ぶ。設計の概念、設計戦略、アーキテクチャ設計、設計の表記法とツール、設計の評価などを扱う。
SE9 ソフトウェア設計 (2) 及び演習	2.0	ソフトウェア設計について学ぶ。ユーザインターフェース設計、SA/SD、JSD、オブジェクト指向設計、デザインパターン、コンポーネントベース設計、インターフェース設計などを扱う。
SE10 ソフトウェア検証 (1) 及び演習	2.0	ソフトウェアテストについて学ぶ。テストの概念、単体テスト、例外の処理、さまざまなカバレッジ、結合テスト、ブラックボックステスト、システムテスト、受け入れテスト、品質特性、回帰テスト、テストツールなどを扱う。
SE11 ソフトウェア検証 (2)	2.0	ソフトウェア検証について学ぶ。検証の基本概念、レビューやインスペクション、ユーザインターフェースのテスト、障害の分析と管理などを扱う。
SE12 開発プロセスと保守	2.0	開発プロセスと保守について学ぶ。プロセスの基本概念、プロセスの種類、品質管理、プロセス改善、規格や標準とテラリング、PSP、TSP、保守の基本概念、保守のプロセス、コストと計画、影響分析、リバースエンジニアリング、リファクタリングなどを扱う。
SE13 ソフトウェア品質	2.0	ソフトウェアの品質について学ぶ。品質の基本概念、社会的意義、品質コスト、品質特性、品質管理、プロセス改善、規格や標準とテラリングなどを扱う。

2.4 SAS(System and Application Specialties)科目

本カリキュラムでは、SAS(System and Application Specialties)科目に 6 単位 101.25 時間を割り当てている。SAS 科目とは、特定の応用分野について学生が深く学ぶための科目である。カリキュラムを立案する立場では、他の学科やコースなどと差別化を図るために設ける応用科目という位置づけになると考えられる。SAS 科目の内容は、学科、コースなどでそれぞれ定めることとし、本カリキュラムでは定めない。参考として IEEE/ACM-CCSE(Computing Curriculum - Software Engineering) で定められている SAS 科目を表 3 に示す。

表 3 IEEE/ACM-CCSE における SAS 科目

科目名
Network-centric systems
Information systems and data processing
Financial and e-commerce systems
Fault tolerant and survivable systems
Highly secure systems
Safety critical systems
Embedded and real-time systems
Biomedical systems
Scientific systems
Telecommunications systems
Avionics and vehicular systems
Industrial process control systems
Multimedia, game and entertainment systems
Systems for small and mobile platforms
Agent-based systems

2.5 演習・卒業研究

本カリキュラムでは、演習・卒業研究に 5 科目 9 単位 247.5 時間を割り当てている。演習・卒業研究の内容を表 4 に示す。チームやプロジェクトでのプログラミング演習は、演習（シーケンシャルプログラミング）や演習（並行プログラミング・オブジェクト指向プログラミング）、もしくは卒業研究 A、卒業研究 B で適宜実施するものとする。ただし情報科学基礎科目及びソフトウェアエンジニアリング科目の一部における演習が付随された講義は含まない。

表 4 演習・卒業研究

科目名	単位数
演習（シーケンシャルプログラミング）	2.0
演習（並行プログラミング・オブジェクト指向プログラミング）	2.0
専門職業演習	1.0
卒業研究 A	2.0
卒業研究 B	2.0

3. 国内外のアクレディテーション基準やカリキュラム標準との関係

本カリキュラムは、JABEE（日本技術者教育認定機構）[1]、ABET-EAC（Accreditation Board for Engineering and Technology：アメリカ）のアクレディテーション基準[2]、及びIEEE/ACM-CCSEのカリキュラム標準[3]に準拠している。以下本章では、各アクレディテーション基準やカリキュラム標準との関係について述べる。

3.1 JABEE との関係

JABEE では、「日本技術者教育認定制度～認定・審査の手順と方法」[1]において6つの認定基準を定めている。そのうち本カリキュラムに関係する「基準 2：学習・教育の量」では、総単位数及び学習時間を以下のように定めている。

- (1) プログラムは4年間に相当する学習・教育で構成され、124単位以上を取得し、学士の学位を得た者を修了生としているか。
- (2) プログラムは学習保証時間（教員の教授・指導のもとに行った学習時間。）の総計が1,800時間以上を有しているか。さらに、その中には、人文科学、社会科学等（語学教育を含む。）の学習250時間以上、数学、自然科学、情報技術の学習250時間以上、および専門分野の学習900時間以上を含んでいるか。なお、ここでいう1時間は、正味の60分とする。

本カリキュラムは、128単位を総単位数とする4年間のカリキュラムであり、修了した者には学士の学位を与えることを想定している。したがって、上記(1)の基準について準拠することになる。

また本カリキュラムの学習保証時間の総計は1800時間であり、2.1節にあるように人文科学・社会科学（語学教育を含む）に258.75時間、数学及び自然科学に303.75時間を割り当てることとなっている。また専門分野の学習として、情報科学基礎科目に517.5時間、ソフトウェアエンジニアリング科目に371.25時間、SAS科目に101.25時間の計990時間を割り当てている。したがって、上記(2)の基準についても準拠することとなる。

以上より、本カリキュラムはJABEEの定める基準に準拠することとなる。

3.2 ABET-EAC との関係

本カリキュラムはABETの定める3つのカテゴリのうち、エンジニアリングに関するカテゴリであるEAC（Engineering Accreditation Commission）の定める基準に従う。ABET-EACでは、"Criteria For Accrediting Engineering Programs"[2]において8つの認定基準を定めている。そのうち本カリキュラムに関係する"Criterion 4. Professional Component"では、総単位数及び学習時間を以下のように定めている。

- (a) *one year of a combination of college level mathematics and basic sciences (some with experimental experience) appropriate to the discipline*
- (b) *one and one-half years of engineering topics, consisting of engineering sciences and engineering design appropriate to the student's field of study*

ABET-EAC では 1 年間の単位数を定めていないため、JABEE で定めている 124 単位を 4 等分した 31 単位を 1 年間の単位数の下限とみなすこととする。すなわち(a)の基準は大まかに「数学及び自然科学を 31 単位以上」となり、(b)の基準は「専門科目を 47 単位以上」と解釈することができる。本カリキュラムは、数学及び自然科学について教養科目で 27 単位、情報科学基礎科目のうち「確率・統計」及び「離散数学と代数構造」の 2 科目で 4 単位の計 31 単位を割り当てることとなっている。したがって、上記(a)の基準について準拠することになる。

また本カリキュラムは専門科目として、情報科学基礎科目（「確率・統計」及び「離散数学と代数構造」を除く）で 33 単位、ソフトウェアエンジニアリング科目で 26 単位、SAS 科目で 6 単位の計 65 単位を割り当てることとなっている。したがって、上記(b)の基準についても準拠することになる。以上より、本カリキュラムは ABET-EAC の定める基準に準拠することとなる。

3.3 IEEE/ACM-CCSE との関係

IEEE と ACM は Joint Task Force を組織し、CCSE (Computing Curriculum - Software Engineering) としてソフトウェアエンジニアリングのカリキュラム標準を整備している。

CCSE の構造は 3 階層である。表 5 に示す 10 の Knowledge Area (以下 KA) と呼ばれるカテゴリから構成され、各カテゴリはさらに複数の Unit と呼ばれるサブカテゴリから構成されている。さらに各 Unit は複数の Topic から構成される。例えば Computing Essentials (略称: CMP) という KA は、Computer Science foundations (略称: CMP.cf) などの Unit から構成される。CMP.cf は、Programming Fundamentals (略称: CMP.cf.1) などの Topic から構成される。

各 Topic には Essential、Desirable、Optional の 3 つのレベルが定められている。Essential と定められた Topic は必ず扱わなくてはならない。Desirable と定められた Topic は、可能であれば扱うことが望ましい。Optional と定められた Topic は、選択科目として扱って構わない。

各 KA 及び Unit には、Hours と呼ばれる学習保証時間が定められている。例えば CMP の Hours は 172 時間であり、CMP.cf の Hours は 140 時間である。各 Topic の Hours は、各学科やコースで適切に割り振る必要がある。CCSE に適合するためには、適切な学習保証時間で各 Topic を扱う必要がある。

本カリキュラムは、各 KA、Unit、Topic について、定められた Hours に従って網羅されている。すなわち、KA、Unit、Topic ごとに扱う科目が定められており、かつ各科目で扱う KA、Unit、Topic の Hours を合計した学習時間が各科目の学習保証時間内であるように割り当てられている。詳細は附表 (IPSJ-CCSE 対応表) を参照のこと。

表 5 IEEE/SEEK における Knowledge Area

Knowledge Area	Hours
Computing Essentials	172
Mathematical and Engineering Fundamentals	89
Professional Practice	35
Software Modeling and Analysis	53
Software Design	45
Software Verification and Validation	42
Software Evolution	10
Software Process	13
Software Quality	16
Software Management	19

4. カリキュラム設置の際のポイント

本カリキュラムは、3 章に示した各学科やコースの特色を表すように設置することが可能である。以下本章では、そのポイントについて述べる。

4.1 総単位数

本カリキュラムの総単位数は 128 単位となっている。したがって、最大 12 単位まで各学科やコースなどの特色を表す科目を増やすことができる。

4.2 SAS 科目

SAS 科目の 6 単位については、各学科やコースなどの特色を表す科目を割り当てることができる。

4.3 情報科学基礎科目

情報科学基礎科目のうち、「情報理論及び符号理論」は CCSE で対応する Topic が存在しない。また「電子回路及び論理回路」「コンピュータアーキテクチャ」「コンパイラ及び演習」ないし「コンピュータ入門」「信頼性工学」は CCSE で対応する Topic が存在するものの、必要とされている Hours が少ない。したがって、これらの科目を扱う必要がないと判断したならば、CCSE で定められている KA/Unit/Topic 及び Hours を満たすことを担保した上で、各学科やコースなどの特色を表す他の科目を割り当てることができる。

4.4 専門職業演習

専門職業演習は、学科やコースなどの特色を表すために、必要ならば工場実習やインターンシップという形態で実施することができる。

5. おわりに

この文書では、JABEE、ABET-EAC、IEEE/ACM-CCSE に準拠した推奨カリキュラムについて概説し、各学科やコースなどで設置する際に特色を表す際のポイントについて解説した。JABEE、ABET-EAC、IEEE/ACM-CCSE の意義を十分理解したうえで、学科やコースなどの特色が表れ、かつ日本のソフトウェア産業を支える学生を多く輩出できるカリキュラムを立案する際の一助となれば幸いである。

参考文献

- [1] 日本技術者教育認定制度～認定・審査の手順と方法：2003 年度適用，日本技術者教育認定機構（2003）.
- [2] ABET Board of Directors, Criteria For Accrediting Engineering Programs : Effective for Evaluations During the 2003-2004 Accreditation Cycle, Engineering Accreditation Commission / Accreditation Board for Engineering and Technology（2002）.
- [3] The Joint Task Force on Computing Curricula, Computing Curriculum ~ Software Engineering : Public Draft 1, IEEE Computer Society / Association for Computing Machinery（2003）.

附表

- IPSJ-SE 教育・訓練カリキュラム表(PDF:16K)
- IPSJ-CCSE 対応表(PDF:229K)

IPJSJ-SE教育・訓練カリキュラム表

開講 学期	和文科目名	欧文科目名	科目説明	単位数		時間数	
				128.0単位	1800.0単位	1800.0単位	1800.0単位
1 年前期	一般教養	General Educations	/	25.0	28.5	281.3	337.5
	コンピュータ入門	Introduction to computers	システムの考え方とコンピュータの仕組みについて学ぶ。計算機内部での数値やデータの表現、CPUやメモリ、I/Oの仕組み、キャッシュや仮想メモリ、バスやDMA、RAIDなどを扱う。	2.0		22.5	
	プログラミング及び演習(1)	Programming with exercise(1)	プログラミングの基礎について学ぶ。高級言語の基本的な文法や変数の型、分岐やループなどの制御構造、デバッグの考え方、基本的なデータ構造などを扱う。また演習を行う。	1.5		33.8	
1 年後期	一般教養	General Educations	/	25.0	30.5	281.3	360.0
	離散数学と代数構造	Discreet mathematics and algebraic structures	離散数学と代数構造について学ぶ。関数や写像、集合論、グラフ理論、代数構造などを扱う。	2.0		22.5	
	プログラミング及び演習(2)	Programming with exercise(2)	プログラミングの基礎について学ぶ。ポインタ、再帰、イベントドリブン、コードの再利用、パラメータ化、防衛的プログラミング、エラー処理、ツールの使い方などを扱う。また演習を行う。	1.5		33.8	
	データ構造とアルゴリズム	Data structure and algorithm	データ構造とアルゴリズムについて学ぶ。アルゴリズムの複雑さやO記法、アルゴリズムの解析、分割統治といったアルゴリズムの考え方、さまざまなアルゴリズム、カプセル化や階層構造といった抽象化などを扱う。	2.0		22.5	
2 年前期	電子回路及び論理回路	Theory of electronic circuits and logic circuits	電子回路、論理回路とデジタル回路、集積回路について学ぶ。コンピュータの構造の進化、ブール代数、論理ゲートやフリップフロップ、順序回路、LSIの仕組みなど組み込みシステムにおけるコ	2.0	18.0	22.5	247.5
	論理と推論	Logic and inference	論理と推論について学ぶ。命題論理、述語論理、論証、演繹と帰納、ホア論理などを扱う。	2.0		22.5	
	確率・統計	Probability and statistics	確率と統計について学ぶ。確率の基礎、分布、統計の基礎、検定と推定、相関、回帰分析ベイズ理論、精度と誤差などを扱う。	2.0		22.5	
	情報理論及び符号理論	Information theory and coding theory	情報理論と符号理論について学ぶ。情報量とエントロピー、情報源符号化、ハフマン符号、伝送情報量と通信容量、通信路符号化定理などを扱う。	2.0		22.5	
	コンピュータアーキテクチャ	Computer architecture	コンピュータのアーキテクチャについて学ぶ。CPUの構造、命令セット、フェッチとデコード、パイプライン、マイクロカーネル、CISCとRISC、キャッシュと仮想記憶、並列プロセッサなどを扱う。	2.0		22.5	
	オペレーティングシステム及び演習	Operating systems with exercise	オペレーティングシステムについて学ぶ。オペレーティングシステムの目的や役割、構造、APIの考え方、割り込み、メモリ保護、並列やデッドロック、スケジューリング、メモリ管理やページングなどを扱う。	2.0		33.8	
	プログラミング言語	Programming languages	プログラミング言語について学ぶ。プログラミング言語の歴史やパラダイム、仮想マシン、インタプリタとコンパイラ、型チェック、ガベージコレクション、抽象化の機構などを扱う。	2.0		22.5	
	デジタル通信とコンピュータネットワーク及び演習	Digital telecommunication and computer network with exercise	デジタル通信とコンピュータネットワークについて学ぶ。プロトコルの基礎、TCP/IP、クライアント・サーバ、Webサービス、分散オブジェクト、暗号化やセキュリティ、データ圧縮、モバイルコンピューティングなどを扱う。	2.0		45.0	
	SE1 構築技術とツール及び演習	Software Engineering(1) construction technology and tools with exercise	構築技術とツールについて学ぶ。構築技術やツール、設計の基本概念、コンポーネントの設計などを扱う。またツールを用いた演習を行う。	2.0		33.8	

IPJSJ-SE教育・訓練カリキュラム表

開講 学期	和文科目名	欧文科目名	科目説明	単位数		時間数	
				128.0単位	1800.0単位	128.0単位	1800.0単位
2 年後期	演習（シーケンシャルプログラミング）	Exercise, training and projects(sequential programming)	プログラミングの演習を行う。逐次処理のアルゴリズムを用いて、個人やチームでプログラミングを行う。	2.0	14.0	67.5	258.8
	形式言語と計算理論	Formal language and theory of computing	形式言語、オートマトン、正規表現および文法、計算理論について学ぶ。	2.0		22.5	
	コンパイラ及び演習	Compilers with exercise	コンパイラについて学ぶ。コンパイラの構造、BNF、字句解析、構文解析、意味解析、誤りの処理、実行時記憶域、コード生成、最適化などを扱う。	2.0		45.0	
	HCIとヒューマンファクター	HCI, and human factors	HCIとヒューマンファクターについて学ぶ。I/Oやエラーメッセージ、GUIといったユーザ側のヒューマンファクターや、コメントやインデント、可読性といった開発者側のヒューマンファクターなどを扱う。	2.0		22.5	
	SE7 ソフトウェアモデル及び演習	Software Engineering (7) software modeling with exercise	ソフトウェアモデリングについて学ぶ。モデリングの基本原則、さまざまなモデル、要求の評価、要求の管理などを扱う。	2.0		33.8	
	SE8 ソフトウェア設計(1) 及び演習	Software Engineering (8) software design with exercise (1)	ソフトウェア設計について学ぶ。設計の概念、設計戦略、アーキテクチャ設計、設計の表記法とツール、設計の評価などを扱う。	2.0		33.8	
	データベース基礎及び演習	Database basics with exercise	データベースの基礎について学ぶ。DBMSの目的と役割、データモデルの紹介、RDBMS、正規化、SQL、トランザクション、物理設計と検索などを扱う。	2.0		33.8	
3 年前期	信頼性工学	Reliability, safety and dependability	信頼性工学について学ぶ。信頼性の基礎、寿命分布と故障率曲線、フォールトトレラント、フェイルセーフ、冗長設計、保全性設計、安全性などを扱う。	2.0	17.0	22.5	247.5
	専門職業演習	Exercise, training and projects(engineering professionalism)	講義と、企業での実習や大学での演習を行う。講義ではグループダイナミクス、人間の心理、コミュニケーション、認証や資格、倫理綱領、技術によって社会的・法的に発生する問題、技術者コミュニティの役割、標準や規格の意義、知的財産権などを扱う。	1.0		22.5	
	SE3 工学基礎	Software Engineering (3) Engineering foundations for software	工学基礎について学ぶ。実験的手法、統計解析、能力評価、セキュリティ、スケーラビリティ、工学的設計の基本概念、ソフトウェア工学以外の分野における工学などを扱う。	2.0		22.5	
	SE2 形式手法とモデル論	Software Engineering (2) formal construction method and modeling foundations	形式手法とモデル論について学ぶ。仕様記述言語、正当性検証、モデリングの概念、事前条件や事後条件などを扱う。	2.0		22.5	
	SE5 組織論とマネジメント	Software Engineering (5) Group dynamics and management	組織論とマネジメントについて学ぶ。グループダイナミクス、人間の心理、マネジメントの基本概念、プロジェクトマネジメント、構成管理などを扱う。	2.0		22.5	
	SE9 ソフトウェア設計(2) 及び演習	Software Engineering (9) software design with exercise (2)	ソフトウェア設計について学ぶ。ユーザインターフェース設計、SA/SD、JSD、オブジェクト指向設計、デザインパターン、コンポーネントベース設計、インターフェース設計などを扱う。	2.0		33.8	
	SAS(System and Application Specialties) 及び演習	SAS(System and Application Specialties) with exercise	制御、AI、ORなど各大学で選択した適用領域について講義する。	4.0		67.5	
	SE10 ソフトウェア検証(1) 及び演習	Software Engineering (10) V&V with exercise (1)	ソフトウェアテストについて学ぶ。テストの概念、単体テスト、例外の処理、さまざまなカバレッジ、結合テスト、ブラックボックステスト、システムテスト、受け入れテスト、品質特性、回帰テスト、テストツールなどを扱う。	2.0		33.8	

IPJSJ-SE教育・訓練カリキュラム表

開講 学期	和文科目名	欧文科目名	科目説明	単位数		時間数	
				128.0単位	1800.0単位		
3 年後期	演習（並行プログラミング・オブジェクト指向プログラミング）	Exercise, training and projects(concurrent programming / object-oriented programming)	プログラミングの演習を行う。並行処理のアルゴリズムおよびオブジェクト指向言語を用いて、個人やチームでプログラミングを組む。	2.0		67.5	
	SE11 ソフトウェア検証（2）	Software Engineering (11) V&V (2)	ソフトウェア検証について学ぶ。検証の基本概念、レビューやインスペクション、ユーザインターフェースのテスト、障害の分析と管理などを扱う。	2.0		22.5	
	SE13 ソフトウェア品質	Software Engineering (13) Quality	ソフトウェアの品質について学ぶ。品質の基本概念、社会的意義、品質コスト、品質特性、品質管理、プロセス改善、規格や標準とテラリングなどを扱う。	2.0		22.5	
	SE4 エンジニアリングエコノミクス	Software Engineering (4) Engineering economics	エンジニアリングエコノミクスについて学ぶ。ライフサイクルにおける価値の変化、参加型デザインやステークホルダー分析、QFDなどによる製品企画、費用対効果分析、価値評価、経済的な影響などを扱う。	2.0	12.0	22.5	202.5
	SAS(System and Application Specialties)及び演習	SAS(System and Application Specialties)with exercise	制御、AI、ORなど各大学で選択した適用領域について講義する。				
	SE12 開発プロセスと保守	Software Engineering (12) Process and Evolution	開発プロセスと保守について学ぶ。プロセスの基本概念、プロセスの種類、品質管理、プロセス改善、規格や標準とテラリング、PSP、TSP、保守の基本概念、保守のプロセス、コストと計画、影響分析、リバースエンジニアリング、リファクタリングなどを扱う。	2.0		33.8	
4 年前期	卒業研究 A	Capstone project A	/	2.0		45.0	
	システム工学と問題解決法	Systems engineering and problem solving	システム工学と問題解決法について学ぶ。システム思考、システムズアプローチなどを扱う。	2.0		22.5	
	SE6 コミュニケーションスキル及び演習	Software Engineering (6) communication skill with exercise	コミュニケーションと要求獲得の技術について学ぶ。コミュニケーションスキル、インタビュー、ブレーンストーミング、プロトタイピング、仕様化とドキュメンテーション、表現、折衝などを扱う。また演習を行う。	2.0	6.0	33.8	101.3
4 年後期	卒業研究 B	Capstone project B	/	2.0	2.0	45.0	