

シラバス表示

シラバスの詳細な内容を表示します。

[→ 閉じる \(シラバスの一覧にもどる\)](#)

科目の基本情報

開講年度	2020 年度
開講区分	工学部情報工学科／総合工学科情報工学コース ・ 専門教育
受講対象学生	学部(学士課程)：2年次 2018年度以前入学者が対象
選択・必修	必修
授業科目名	上級プログラミング演習 I・II (再履修) じょうきゅうぷろぐらみんぐえんしゅう 1・2 Advanced Programming Exercise I & II
単位数	各1 単位
ナンバリングコード	EN-CMPS-2

開放科目	非開放科目
開講学期	通年 I を前期に，II を後期に開講
開講時間	前期は木曜日3,4時限，後期は月曜日7,8時限
開講場所	情報棟 電算演習室
担当教員	各教員 各テーマの担当教員，2年担任
SDGsの目標	

学修の目的と方法

授業の概要	基礎および中級で習得したCプログラミング技術を，実際の問題に応用する方法を修得することを目的とする．他の授業科目で習うアルゴリズムなどを題材にした問題に対して，問題の分析，処理手順の設計，Cプログラムによる実装を行う．
学修の目的	与えられた問題に対して，その問題の分析，処理手順の設計，Cプログラムによる実装ができるようになる．
学修の到達目標	与えられた問題に対して，その問題の分析，処理手順の設計，Cプログラムによる実装ができるようになる．
	<ul style="list-style-type: none"> ○ 学科・コース等の教育目標 ○ JABEE 関連項目 <ul style="list-style-type: none"> 情報工学の基礎知識を身につけ、科学技術が社会や自然環境に及ぼす影響を理

ディプロマ・ポリシー

解し、責任ある技術者として行動できる。【技術者倫理】

情報工学科に関連する様々な分野に関心を持ち、未知分野を理解するために、自主的、継続的に学習できる。【自主的継続的学習】

世界に多様な考え方があることを学び、様々な立場の考えや意見を尊重し、多面的に物事を考えることができる。

【多面的な思考能力】

情報工学に関連する課題に対して、与えられた条件や期限を熟慮し、計画的に作業を進め、報告できる。【計画的な活動】

専門分野の英語で書かれた文献について理解し、説明できる。また、学習や実験で得た知見を、論理的に記述し、的確に発表し、討議できる。【コミュニケーション能力】

自然科学と情報技術に関する十分な知識を修得し、それらの知識を応用できる。【知識の修得と応用】

与えられた問題に対し、修得した知識や技術を利用して、関連情報を収集し、解決手法を提案し、実現できる。【問題解決能力】

○ 全学の教育目標

感じる力

感性 共感 ○主体性

考える力

幅広い教養 ○専門知識・技術 ○論理的・批判的思考力

コミュニケーション力

表現力(発表・討論・対話) リーダーシップ・フォロワーシップ 実践外国語力

生きる力

	○問題発見解決力 心身・健康に対する意識 社会人としての態度・倫理観
成績評価方法と基準	全テーマの報告書を出した者だけに単位を与える。 評価は、各テーマ担当教員が内容等を考慮して付けた点数の総計を10点満点に換算し、6以上を合格とする。 単位は、前期に「上級プログラミング演習Ⅰ」として1単位、後期に「上級プログラミング演習Ⅱ」として1単位、個別に認定する。
授業の方法	演習
授業の特徴	問題提示型PBL（事例シナリオ活用含） 問題自己設定型PBL プロジェクト型PBL 実地体験型PBL
授業改善の工夫	授業アンケート等で寄せられた受講者からの意見を反映し、演習内容や演習方法を改善する。
教科書	教科書：指導書として演習ウェブページが用意される。
参考書	参考書：演習ウェブページで指示する。
オフィスアワー	演習実施時に、各テーマ担当の教員が指示する。
受講要件	
予め履修が望ましい科目	「プログラミング基礎及び演習」及び「中級プログラミング及び演習」の単位修得が強く望まれる。習得していないと履修できない訳ではないが、現実的には演習指導書に記載のアルゴリズムをプログラムとして記述できないので、非常に苦勞し、結果的には（今までの経験則によると96%以上の確率で）合格しない。また、各テーマに関連する科目が演習の実施中に開講されるの

	で、それらを履修することが望ましい。
発展科目	情報工学実験 I, II 上級プログラミング演習 III 卒業研究
その他	再履修にあたっては、前期の履修登録時に担任と科目担当教員に連絡し指示に従うこと。

授業計画

MoodleのコースURL

キーワード	基礎技術, デバッグ, ソート, 画像処理, データ管理, 数式処理, 配列, リスト構造, 木構造, TeX, Make, tcsh, gnuplot, GDB, GCC
Key Word(s)	debug, sort, graphics processing, data management, formula manipulation, array, list, tree, TeX, Make, tcsh, gnuplot, GDB, GCC
	<p>前期と後期に3テーマずつ、次の順で演習を実施する。</p> <p>1. 基礎技術（5回） 上級プログラミング演習の実施に必要な基礎的な技術の習得を目的とする。 具体的には、以下の入門的な実習を行う。</p> <ul style="list-style-type: none"> ・組版ソフトウェア TeX の使用法 ・プログラミングの効率化のための make コマンド ・コマンドインタプリタである tcsh の使用法およびスクリプトの作成法 <p>2. デバッグ技術（5回） 誤りのないプログラムを作成する、あるいは誤りを効率よく見つけるために、以下の技術を習得する。</p> <ul style="list-style-type: none"> ・コンパイル時のエラーメッセージの見方と対処法 ・誤りを生じにくいコーディング手法 ・デバッガの使い方

学修内
容

3. ソートとその応用 (5回)

ソート (ソーティング) に関するプログラミング課題に取り組むことにより,

- ・ 問題分析やプログラム設計の能力を養い,
- ・ データ構造やアルゴリズムの重要性を認識し,
- ・ プログラム作成の技術を高める.

また, gnuplotを用いて実験結果をグラフ化する手法を習得する.

4. 配列と画像処理 (5回)

画像処理プログラムの作成を通して, 以下の事柄を理解し習得する.

- ・ 計算機を用いた画像処理, 解析手法の基礎
- ・ メモリの動的確保など, C言語における動的メモリ処理の方法
- ・ 画像ファイルに代表されるバイナリ形式ファイルの入出力
- ・ 拡張性や再利用性を考慮したプログラミング

5. リスト構造によるデータ管理 (5回)

リスト構造を使ったデータベースシステムを構築することにより,

- ・ 「リスト構造」というデータ構造の考え方と操作方法を理解し,
- ・ システム構築の実装感覚を養う.

6. 木構造と数式処理 (5回)

木構造を利用した数式処理のプログラミング課題に取り組むことにより,

- ・ 問題分析やプログラム設計の能力を養い,
- ・ データ構造やアルゴリズムの重要性を認識し,
- ・ プログラム作成の技術を高める.

各テーマの最終週は, まとめの週とする. 最終週には演習室で自習し, 演習内容の復習, 未完の課題や発展課題の実施, 考察, 報告書の準備などにあたる. (ただし, テーマ2は内容が多いため, 通常の演習をする.)

事前・ 事後学 修の内 容	事前に実験内容を予習して実験手順を把握し、当日に演習を円滑に進めるための準備をしておく。 詳細については、テーマごとの演習ウェブページ等を読む。
----------------------------------	---

[→シラバス執筆の手引き](#) | [→シラバストップ](#)

Copyright (c) Mie University