

1. 科目コード

1208

2. 科目名

プログラミング基礎論 (Fundamentals of Computer Programing)

3. 担当教員

大寺 亮 (Ryo OHTERA)

4. 開講期

春1期 ((昼) 火曜 3-4 時限、(夜) 火曜 6-7 時限)

5. 科目の目的・概要

プログラミングは、本学で IT について学ぶ上で基礎となる知識・技術である。これまでに、一度もプログラミングを学んだことがない学生にとっては、本講義が最初の一步となる。本講義は、Linux を用いた、C 言語によるプログラミングの基礎の習得ならびにプログラム作成に関するスキルの定着を目指す。そのために、演習により実際に手を動かしながら、プログラミングを学ぶ。

6. 科目の学習目標

- (1) C 言語の基礎文法を理解し、作成したプログラムを実行できる
- (2) C 言語プログラムにおける処理の流れを把握し、100 行程度のプログラムの内容を理解できる
- (3) プログラムとして達成したい事について、学んだ基礎知識をもとに、全体の流れを自身で考える事ができる
- (4) 学んだ基礎知識に若干の応用を加える事ができる

7. 本学の教育目標と科目の学習目標との対応

教育目標			学習目標
高度 ICT スキルの修得	基礎的素養		(1), (2), (3)
	専門知識および業務応用力		
人間力（＝探究力）の修得	自ら強みを磨き続ける力		
	自ら社会における課題を発見し、解決する力	課題設定	
		仮説立案	
		仮説検証	
		実行	
	社会人基礎力	前に踏出す力	(4)
		考え抜く力	(3), (4)
		チームで働く力	
職業倫理の修得			

8. 履修要件

- (1) 『コンピュータシステム基礎論』を受講していることが望ましい

9. 教科書

なし

10. 参考書

なし

11. 評価方法と配点

学習目標	達成度評価方法と配点				
	期末試験	小テスト	レポート	発表	成果物 (授業中演習課題)
(1)	○		○		
(2)	○		○		
(3)					○
(4)					○
配点	60		30		10

12. 備考

本授業では教科書を利用しないが、授業資料を Moodle にて公開する

本授業は、入学前特別集中講義の『Linux&C 言語プログラミング演習』を受講し、Vmware で Linux 環境を構築している事、Linux の初歩的な使用法を学んでいる事、および vi の基本操作を学んでいる事を前提とする。

■ 授業計画

(注)授業計画は、あくまでも予定であり、実施時に、適時、追加・変更・修正等が生じる場合があります。

第1回 プログラム基礎

(講義 45 分、演習 45 分)

本講義をはじめるにあたり、オリエンテーションを行う。また、適宜演習をしながら、C言語によるプログラミングの準備を行う。

1. オリエンテーション
 - (ア) 講義の流れ
 - (イ) 学習目標
 - (ウ) 評価の方法
 - (エ) 評価基準
2. プログラミングの流れ
3. Linux コマンド基礎練習
4. Vi 基礎練習
5. C 言語プログラムの構成
6. プログラムの実行
7. printf の使い方
8. 目標達成度確認のための演習問題

第2回 データ構造①

(講義 45 分、演習 45 分)

C 言語プログラミングにおいて重要な変数の概念について、演習を行いながら数値演算を例に説明する。

1. 変数とは何か
2. 変数の操作
 - (ア) 値の設定
 - (イ) 値の参照
3. 変数の定義
4. 変数の型
5. 算術・代入演算子
6. 標準出力・標準入力
7. 目標達成度確認のための演習問題
8. ミニレポート

第3回 処理構造①(選択)

(講義 45 分、演習 45 分)

演習を行いながら条件に応じて処理を選択する if 文について説明する。それに伴い、条件の概念や条件を記述するための演算子について説明する。

1. 応用復習問題
2. 明示的型変換
3. if 文と if～else 文
4. 関係演算子
5. 論理式と論理演算子
6. switch～case 文
7. break
8. 入れ子構造
9. 目標達成度確認のための演習問題

第4回 データ構造①、処理構造① 演習**(演習 90 分)**

変数の取り扱いと条件に応じて処理を選択する処理を、自ら考えながらプログラミングできるよう、演習を繰り返す。

1. データ構造(変数)演習
2. 選択処理演習
3. ミニレポート

第5回 処理構造②(反復)**(講義 45 分、演習 45 分)**

処理を繰り返す反復の構造を学ぶために、演習を行いながら for 文やその他の反復法について説明する。また、読みやすいプログラムを書くための作法についても説明する。

1. while 文と do～while 文
2. 永久ループ
3. for 文
4. while 文と for 文の比較
5. インクリメントとデクリメント
6. 複合代入演算子
7. プログラミングの作法
 - (ア) コメント
 - (イ) インデント
 - (ウ) 桁位置合わせ
 - (エ) 空白
8. 目標達成度確認のための演習問題

第6回 処理構造②(反復)演習**(演習 90 分)**

処理を繰り返す反復についてを、自ら考えながらプログラミングできるよう、演習を繰り返す。

1. 反復処理演習
2. ミニレポート

第7回 データ構造②(配列)**(講義 45 分、演習 45 分)**

複数の値を扱うために、演習を行いながら配列(1次元&2次元)について説明する。

1. 配列とは何か
2. 配列の定義
3. 変数と配列の違い
4. 配列の入出力
5. 2次元配列とは何か
6. 2次元配列の入出力
7. 2次元配列の入れ子構造
8. 目標達成度確認のための演習問題

第8回 データ構造②(配列)演習**(演習 90 分)**

複数の値を扱うために、配列(1次元&2次元)について説明する。

1. 配列演習

2. ミニレポート

第9回 ポインタ**(講義 45 分、演習 45 分)**

演習を行いながらポインタとアドレスについて説明する。

1. ポインタとは何か
2. メモリ空間とアドレス
3. 変数の復習
4. 変数のアドレスの受け渡し
5. 変数とポインタの違い
6. 配列とポインタ
7. 目標達成度確認のための演習問題

第10回 ポインタ 演習**(演習 90 分)**

ポインタの基礎について、自ら考えながらプログラミングできるよう、演習を繰り返す。

1. ポインタ演習
2. ミニレポート

第11回 関数(関数の定義、引数)**(講義 45 分、演習 45 分)**

プログラムの部品化のため、演習を行いながら関数の定義方法や呼び出し方について説明する。

1. これまでのプログラミングの振り返り
 - (ア) main 関数のみのプログラム
 - (イ) 用意されている関数(例: printf 関数)
2. 引数と戻り値
3. 関数の自作方法
 - (ウ) 関数のプロトタイプ宣言
 - (エ) 関数の定義
 - (オ) 関数の呼び出し
 - (カ) 関数の型
4. スコープルール
 - (キ) グローバル変数
 - (ク) ローカル変数
5. 標準ライブラリ関数
6. 目標達成度確認のための演習問題

第12回 関数演習**(演習 90 分)**

関数について、自ら考えながらプログラミングできるよう、演習を繰り返す。標準ライブラリ関数(続き)。

1. 標準ライブラリ関数
2. 関数演習
3. ミニレポート

第13回 文字列処理**(講義 45 分、演習 45 分)**

数値以外の文字を自在に扱うため、演習を行いながら文字列について説明する。

1. 文字列を扱うための配列とポインタの復習
2. 文字列とは何か
3. 文字と文字列の違い
4. 文字と文字列の型
5. 文字と文字列の参照
6. 文字列を文字の集合と考える場合
7. 文字と文字列の設定まとめ
8. 文字と文字列の参照まとめ
9. 目標達成度確認のための演習問題

第14回 文字列処理 演習

(演習 90 分)

文字列について、自ら考えながらプログラミングできるよう、演習を繰り返す。

1. 文字列処理演習
2. プログラミングコンテスト準備
3. ミニレポート

第15回 プログラミングコンテスト

(講義 45 分、演習 45 分)

プログラミングにおいて、自ら考える力を養うため、与えるテーマに沿ったプログラミングを行う。

1. プログラミングコンテスト

第16回 理解の確認と評価

(講義 90 分)

講義全体に対する理解度の確認として、ペーパーテストを行う。プログラムの穴埋めや、C 言語に関する用語について問う。

1. 試験