木更津工業高等専門学校	開講年度	平成31年度 (2019年度)	授業科目	日本事情Ⅲ

科目基礎情報

科目番号	0115	科目区分	一般 / 必修（留学生）
授業形態	授業	単位の種別と単位数	履修単位: 2
開設学科	電子制御工学科	対象学年	3
開設期	通年	週時間数	2
教科書/教材	内容に合わせて作成した講義資料を配布する		
担当教員	岡本 峰基		

到達目標

・ 基本的な電子部品・計測機器の取り扱いができる。
・実験実習3で使用するラインとレーサーを完成させる。
・本学科を卒業し将来ソフトウェアを作成または利用する業務に携わるための基礎として、プログラミングの基礎的知識と技能を身に付けること。目安として：
□ 処理の順序を、フローチャートで表現することができる。
□ ソースコードにインデントやコメントを用いてプログラムを作成することができる。
□ 基礎的なC言語のプログラムの動作を説明することができる。
□ 構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文を、C言語において正しく記述できる。

ルーブリック

	理想的な到達レベルの目安	標準的な到達レベルの目安	未到達レベルの目安
評価項目1	原理を十分に理解したうえで、製作物であるライントレーサーを製作できる。	大まかな原理を理解したうえで、製作物であるラインとレーサーを製作できる。	製作物であるラインとレーサーを製作できない。
評価項目2	処理の順序を、正しく分かりやすいフローチャートで表現することができる。	処理の順序を、フローチャートで表現することができる。	処理の順序を、フローチャートで表現することができない。
評価項目3	コメントの内容や変数名なども分かりやすいプログラムを作成することができる。	ソースコードにインデントやコメントを用いてプログラムを作成することができる。	ソースコードにインデントやコメントを用いてプログラムを作成することができない。
評価項目4	基礎的なC言語のプログラムの動作を説明することができるだけではなく、改良するための工夫ができる。	基礎的なC言語のプログラムの動作を説明することができる。	基礎的なC言語のプログラムの動作を理解することができない。

学科の到達目標項目との関係

教育方法等

概要	前半:半田付けの練習を行ったうえで、１人１台のライントレーサーの製作を行う。 後半：コンピュータを用いてソフトウェアを作るためのプログラミングの基礎的知識と技能を、C言語を用いて学び演習する。
授業の進め方・方法	前半：実験実習３で使用するラインとレーサーを各自で製作する。ユニバーサル基盤と電子部品を配布するので、回路図どおりにライントレーサーを製作する。 後半：スライドを用いた講義をなるべく手短に受けた後で、演習課題によって学習内容を身に付ける。
注意点	前半:ライントレーサーの製作は、適時、説明をします。説明中は、作業を一時中断して、説明をちゃんと聞くこと。 後半：プログラミングの知識と技能は、実験実習・情報処理・制御工学・卒業研究ほか多くの専門科目の礎の一つである。したがって、分からない個所があれば、積極的に質問して理解すること。学習内容は履修者に合わせて適宜変更することがある。

授業計画

		週	授業内容	週ごとの到達目標
前期	1stQ	1週	ガイダンス	実験の流れを説明できる。
		2週	ライントレーサー製作の準備１	ライントレーサーの構造を理解する。
		3週	ライントレーサー製作の準備２	適切な半田付けが出来る。
		4週	ライントレーサーの製作１	ライントレーサーを製作する。
		5週	ライントレーサーの製作2	ライントレーサーを製作する。
		6週	ライントレーサーの製作３	ライントレーサーを製作する。
		7週	ライントレーサーの製作４	ライントレーサーを製作する。
		8週	ライントレーサーの製作５	ライントレーサーを製作する。
	2ndQ	9週	ライントレーサーの製作６	ライントレーサーを製作する。
		10週	ライントレーサーの製作７	ライントレーサーを製作する。
		11週	ライントレーサーの製作８	ライントレーサーを製作する。
		12週	ライントレーサーの製作９	ライントレーサーを製作する。
		13週	ライントレーサーの製作１０	ライントレーサーを製作する。
		14週	ライントレーサーの製作１１	ライントレーサーを製作する。
		15週	ライントレーサーの製作１２	ライントレーサーを完成させる。
		16週		
後期	3rdQ	1週	【プログラミングの技と法】 「フローチャート」 ○シラバスの説明 ○ガイダンス ○プログラミングの歴史 ○構造化プログラミング ○フローチャート ☆課題（構造化プログラミング・フローチャート）	□プログラミングの技法をなぜ身に付けるべきであるのか考える。 □プログラミングの歴史的経緯を知る。 □【特に重要】プログラムの流れを順次構造、分岐構造、反復構造の組み合わせで表すことができる。 □フローチャートを正しい描き方で描く事ができる。

		2週	「UNIX（入門編）」 ○2進数・10進数・16進数・ASCII文字 ○UNIX環境の操作方法 ○エディタemacsの操作方法 ○簡単なプログラムの作成	□コンピュータで整数や文字がどのように表されているか、知人や家族に説明できる。 □UNIX環境を基本的なコマンドで操作することができる。 □emacsでテキストファイルを編集することができる。 □キーボード操作における初歩的なミス（Caps LockやInsertなど）に対処することができる。
		3週	【順次構造】 「C言語入門」 ○プログラム作成の手順 ○基本的なプログラムの作成 ○コンパイラgccのはたらき ○基本的なプログラムをあえて間違えてみる ○課題の提出方法（メールの使い方） ☆課題（適切な書式、画面表示）	□ソースファイルをコンパイル・デバッグして実行ファイルを作成する手順を実施できる。 □【特に重要】プログラムを適切な書式（インデント・コメント）で書くことができる。 □ソースファイルからどのような仕組みで実行ファイルが作成されるか、理解する。 □プログラムに間違いがあったときにどうなるか、おおまかに知る。
		4週	「整数・文字と計算」 ○整数や文字を表示する方法 ○変数を使って計算する方法 ○整数・文字のデータ型とバイト数 ○変数のもつ値とアドレス ○数値や文字をキーボード入力する方法 ☆課題（printf()、scanf()、変数、整数型、アドレス、sizeof演算子）	□表示したい内容に合わせてprintf()の書式を使い分けることができる。 □キーボード入力したい内容に合わせてscanf()の書式を使い分けることができる。 □整数と文字のデータ型の種類と、表せる範囲を理解する。 □整数型の変数と基礎的な演算子を使ったプログラムを作成することができる。 □【特に重要】メモリ上に割り当てられている変数のアドレス（番地）を求めることができる。
		5週	「浮動小数点数と計算」 ○固定小数点数と浮動小数点数 ○丸め誤差 ○浮動小数点数のデータ型とバイト数 ○実数と整数の割り算にご用心 ○キャスト演算子を用いた型変換 ○定数に名前を付ける（constと#define） ☆課題（printf()、scanf()、変数、浮動小数点数型、定数、整数÷整数）	□浮動小数点数型の変数を用いて、実数を扱うプログラムを作成することができる。 □【特に重要】整数型と浮動小数点数型の違いを正しく理解して扱うことができる。 □キャスト演算子を用いて型変換を行うことができる。 □定数に名前を付ける二つの方法を使うことができる。
		6週	「配列と文字列」 ○配列の定義・初期化・要素・アドレス ○文字列の定義・初期化・要素・アドレス ○2次元配列 ☆課題（文字列、割り算のあまり、整数・文字）	□配列・文字列を用いて複数のメモリをまとめて扱うことができる。 □【特に重要】配列にまつわる「値」と「アドレス」の区別に注意してプログラミングを行うことができる。
		7週	「ポインタ」 ○ポインタ変数（変数のアドレスを扱う変数） ○NULLポインタ ○なぜややこしいポインタを使うのか ☆課題（ポインタ、ポインタの配列、アドレス）	□ポインタを使ったプログラムを作成することができる。 □【特に重要】ポインタにまつわる「値」と「アドレス」の区別に注意してプログラミングを行うことができる。
		8週	「前期中間試験」（予定） ○プログラミング関連の英単語 ○プログラミングに関する基礎用語 ○基本操作 ○デバッグ ○プログラム作成 ○フローチャート	前期中間試験までの内容を説明あるいは実践できる。特に： □処理の順序をフローチャートで表現することができる。 □基礎的なC言語のプログラムの動作を説明することができる。
	4thQ	9週	○前期中間試験の解答と解説	□試験で間違った内容から、何に集中して復習すれば効率的に知識・技術を身に付けることができるか、前向きに取り組むことができる。
		10週	「if-else文」 ○if文・if-else文（条件式による分岐構文） ○条件式 ○関係演算子・等価演算子・論理演算子 ○演算子の優先順位 ○time.hでUNIX時間を扱う ☆課題（if-else文、条件式、素数、★エラストテネスのふるい、割り算のあま	□if-else文による分岐構文を用いたプログラムを作成することができる。 □【特に重要】関係演算子・等価演算子・論理演算子を用いて適切な条件式を記述することができる。 □stdio.h以外のヘッダファイルを用いることで、利用可能な関数が増えることを理解する。 □素数を判定するための基礎的なアルゴリズムの仕組みを説明することができる。
		11週	「switch文」 ○switch文（整数・文字の値による分岐構文） ○if文とswitch文の違い ○stdlib.hで疑似乱数を扱う ○じゃんけんゲームのプログラム ○goto文 ☆課題（switch文、break、default）	□switch文による分岐構文を用いたプログラムを作成することができる。 □goto文の使用が基本的に推奨されない理由を理解する。
		12週	【反復構造】 「while文」 ○while文（条件式による反復構文） ○インクリメント／デクリメント演算子 ○break文とcontinue文 ○do-while文 ☆課題（while文、定数、乱数、UNIX時間、条件式、break、continue、ゲーム）	□while文による反復構文を用いたプログラムを作成することができる。 □break文とcontinue文の違いを理解して使い分けることができる。 □定数、条件式、乱数、UNIX時間など、以前に学習した内容が身についている。
		13週	「for文」 ○for文（C言語でよく使われる反復構文） ○for文と配列 ○for文と浮動小数点数 ○C言語の分岐構文・反復構文のまとめ ☆課題（for文、定数、配列、値とアドレス、★最高・最低・合計・平均を求めるアルゴリズム）	□【特に重要】for文による反復構文を用いたプログラムを作成することができる。 □for文のカウント変数として浮動小数点数を用いてはいけない理由を理解する。 □【特に重要】最高・最低・平均を求めるための基礎的なアルゴリズムの仕組みを説明することができる。
		14週	「数学関数の使用」 ○math.hで数学関数を扱う ○無限（inf）と非数（nan） ☆課題（for文、定数、三角関数、度とラジアン）	□数学関数を用いたプログラムを作成することができる。 □for文を用いて、表に近い形式で計算結果を表示することができる。

		15週	「前期定期試験」（予定） ○プログラミング関連の英単語 ○プログラミングに関する基礎用語 ○条件式の書き方 ○値とアドレスの区別 ○プログラム作成	前期定期試験までの内容を説明あるいは実践できる。 特に： □構造化プログラミングにおける順次構造・分岐構造・反復構造の構造化文を、C言語において正しく記述できる。 □値とアドレスを意識して使い分けることができる。
		16週	○前期定期試験の解答と解説 「プログラミング演習」 ☆課題（for文、if-else文、break文、while文、数学関数、キャスト演算子、switch文）	□試験で間違った内容から、何に集中して復習すれば効率的に知識・技術を身に付けることができるか、前向きに取り組むことができる。 □前回までの学習内容を応用して、for文を用いた数当てゲームのプログラムや、解の公式で2次方程式の解を求めるプログラムや、pHから水素イオン濃度を求めるプログラムを作成することができる。

評価割合

	試験	発表	相互評価	態度	ポートフォリオ	その他	合計
総合評価割合	25	0	0	0	75	0	100
基礎的能力	0	0	0	0	0	0	0
専門的能力	25	0	0	0	75	0	100
分野横断的能力	0	0	0	0	0	0	0