

## 調査 5 モデルカリキュラムの提言 コースウェア

### 14. C、C++に関するスキル

I. 概要	オープンソースシステムのソースを解読する上で、C や C++の知識は必須であるといえる。本カリキュラムでは、まず C の基礎的な構造やポインタ、メモリ管理について理解を深める。次にオブジェクト指向に関する理解を深めるために、C++について学習する。最後に、オープンソースとして公開されているライブラリを利用して、GUI アプリケーションなどの開発技法について学習する。
II. 対象専門分野	職種共通
III. 受講対象者、 受講前提	<ul style="list-style-type: none"> <li>・ プログラミング経験があること</li> </ul>
IV. 学習目標	<ul style="list-style-type: none"> <li>• C の概要や C プログラミングの基本文法に関して理解する。</li> <li>• C++の概要や C++プログラミングの基本文法に関して理解する。</li> <li>• コンソール入出力やファイル入出力に関して理解する。</li> <li>• ポインタやメモリ管理など、C や C++を学ぶ上での重要項目を学習する。</li> <li>• オブジェクト指向プログラミング(クラスとオブジェクトの関係)</li> <li>• C++の持つ継承、カプセル化、オーバーライドなどの優れた機能をとおしてオブジェクト指向型プログラムの理解を目指す。</li> <li>• GUI ライブラリやその他開発ライブラリを利用したプログラムを理解する。</li> </ul>
V. 使用教科書、 教材等	『C 実践プログラミング』 Steve Oualline 著、オライリー・ジャパン刊 『C++実践プログラミング』 Steve Oualline 著、オライリー・ジャパン刊
VI. 習得スキルの 評価方法	講義終了後の受講レポート、定量アンケート、知識確認ミニテスト、演習問題の取り組み状況を総合的に判断して評価を行う。
VII. カリキュラム の構成	レベル 1 第 1 回～第 9 回 レベル 2 第 10 回～第 15 回

## 講座内容

---

### 第1回 Cの基本(講義+ワークショップ 90分)

---

Cプログラミングの概要として、歴史や特徴、適用範囲について学習する。

また、プログラムを開発する流れについて習得する。

#### (1)Cの概要

1. Cの歴史(開発の経緯)
2. Cの特徴(他言語との比較、オープンソースシステムを学習する上での必要性)
3. Cの開発事例(適用範囲)

#### (2)Cによる開発の流れ

1. エディタの基本操作(viエディタ)
  2. コンパイラの導入(gcc)
  3. プログラム実行までの流れ
  4. 簡単なCプログラムの作成
- 

### 第2回 Cの基本構造(講義+ワークショップ 90分)

---

Cの基本的な仕組みと構成、基本構文を学習し、プログラミングの内容を理解する。

また、サンプルプログラムをもとに、Cで作成したプログラムの流れを確認する。

#### (1)Cの書き方の特徴

1. Cのファイル構成(ソースファイル、ヘッダーファイル)
2. Cのプログラム構成(main関数、標準ライブラリ、等)
3. Cの基本文法(書式、予約語、注釈文)

#### (2)基本的なプログラム記述の例

1. 変数、定数
2. データ型
3. 演算子(代入演算子、算術演算子、比較演算子、論理演算子、演算子の優先順位)
4. 型変換(暗黙の型変換、明示的な(キャスト)型変換)
5. 制御構造(条件分岐処理、繰り返し処理)
6. プリプロセッサ
7. サンプルプログラム(標準出力による実行例)

---

### 第3回 文字列操作(講義+ワークショップ 90分)

---

配列の扱いを学習し、Cによる文字列に対する処理方法について習得する。また、文字列操作の関数を使用し、より簡潔なプログラム作成法について学習する。

#### (1) 配列と文字列

1. 配列とは(変数と配列の違い、宣言方法、等)
2. 配列と文字列(配列による文字列の扱い)
3. 2次元配列と多次元配列
4. サンプルプログラム(文字列のコピー)

#### (2) 文字列操作関数

1. 文字列操作関数の概要
2. 文字列操作関数の利用方法
3. 文字列操作関数の種類(strcpy 関数、strlen 関数、等)
4. サンプルプログラムの簡略化((1)のプログラムを改変)

---

### 第4回 関数(講義+ワークショップ 90分)

---

ユーザ定義関数の扱いを学習し、Cプログラムの部品化の方法を習得する。また、前節で学習した文字列操作関数以外の標準関数を利用したプログラミングについて学習する。

#### (1) 関数の説明

1. 関数とは(プログラム部品化の必要性、関数の利点)
2. 関数の種類(ユーザ定義関数、標準関数、マクロ関数)

#### (2) ユーザ定義関数

1. ユーザ定義関数の定義(プロトタイプ宣言、戻り値、引数の設定)
2. 関数の呼び出し(データの受け渡し方法)
3. ユーザ定義関数とスコープ(記憶クラス)
4. サンプルプログラム

#### (3) 標準関数

1. 標準関数(時刻・日付、文字処理、算術処理、メモリ管理)の利用方法
2. 標準関数の種類(ヘッダーごとに利用できる関数群の説明)
3. サンプルプログラム

---

## 第5回 ポインタ(講義+ワークショップ 90分)

---

ポインタの扱いを学習し、アドレスによる値の参照方法について習得する。  
また、ポインタを利用した文字列、配列、関数の扱いについて学習する。

### (1)ポインタの基礎

1. ポインタとは(アドレスの概念、メモリの概念)
2. ポインタ変数(ポインタ変数の宣言、アドレスの設定、取得)
3. サンプルプログラム(アドレスの取得方法、値の参照方法)

### (2)ポインタによるプログラミング

1. ポインタと配列(ポインタのポインタ)
2. ポインタと文字列
3. ポインタと関数(引数の値を参照する方法、引数に配列を指定する方法)
4. 関数ポインタ(関数ポインタによる関数の呼び出し)
5. サンプルプログラム

---

## 第6回 構造体(講義+ワークショップ 90分)

---

構造体の扱いを学習し、変数や配列をまとめて扱う方法について習得する。  
また、構造体を使用したプログラムについて学習する。

### (1)構造体の概要

1. 構造体とは(変数や配列によるデータ管理の違い)
2. 構造体の定義(定義方法、メンバ変数(配列)の指定、構造体の入れ子構造)
3. 構造体の使用(宣言方法、データの初期化、メンバ変数へのアクセス方法)

### (2)構造体を使用したプログラム

1. 構造体と配列(構造体配列の宣言、初期化)
2. 構造体とポインタ(メンバ変数へのアクセス方法)
3. 構造体と関数(引数に構造体を指定する方法)
4. サンプルプログラム

---

## 第7回 コンソール入出力(講義+ワークショップ 90分)

---

標準入出力や、対話型入出力について学習し、キーボードで入力した情報をプログラムに渡し、処理した後に表示する方法について学習する。

### (1) コンソール入出力の説明

1. コンソール入出力とは
2. コンソール入出力の種類(標準出力、対話型出力)

### (2) 標準入出力

1. 文字の入出力(getchar/putchar)
2. 文字列の入出力(gets/puts)
3. 書式付き入出力(scanf/printf)と書式指定
4. コマンドライン引数(パイプによる値渡し)
5. サンプルプログラム

### (3) 対話型入出力

1. 対話型入出力とは(対話型入出力を実現するライブラリ(curses)の説明)
2. cursesの基本構成(ライブラリの導入、ライブラリ内の関数群、ライブラリを指定したコンパイル)
3. 対話型入出力の処理の流れ(プログラムの流れ)
4. サンプルプログラム
5. cursesライブラリを使用した応用例(プログラム例の紹介)

---

## 第8回 ファイル管理(講義+ワークショップ 90分)

---

高水準入出力関数と低水準入出力関数について学習し、ファイルにアクセスして、データを入出力する方法について習得する。また、ファイル操作やディレクトリ操作といった、ファイルを管理する上で必要となるプログラムについて学習する。

### (1) ファイル管理の説明

1. ファイル管理とは
2. ファイル管理の種類(ファイル入出力、ファイル操作、ディレクトリ操作)

### (2) ファイル入出力

1. 高水準ファイル入出力(FILE 構造体)
2. 低水準ファイル入出力(ファイルディスクリプタ)
3. 高水準ファイル入出力と低水準ファイル入出力の違い
4. ファイル入出力の流れ(ファイルオープン、ファイルクローズ、入出力処理)
5. ファイルエラーとランダムアクセス
6. サンプルプログラム

### (3) ファイル操作とディレクトリ操作

1. ファイル操作(ファイル検索、ファイル削除、ファイル名変更)
2. ディレクトリ操作(ディレクトリ作成、ディレクトリ削除)
3. サンプルプログラム

---

## 第9回 データ構造(講義+ワークショップ 90分)

---

データを格納する仕組みを定義したデータ構造について学習し、用途に応じたデータ構造の使用方法を習得する。また、データ構造を使用したプログラムについて学習する。

### (1) データ構造の説明

1. データ構造とは(概要、アルゴリズム、計算量、等)
2. データ構造の種類(線形リスト、木、スタック、キュー、等)
3. データ構造とメモリ管理(動的なデータ領域の確保、sizeof 演算子、malloc 関数、free 関数)

### (2) 線形リストとは(データ構造の一例)

1. 線形リストの概要
2. 線形リストの種類(単方向リスト、双方向リスト、環状リスト)と適用例
3. 線形リストによるプログラム例(線形リストの構築例、解説)
4. サンプルプログラム(3のプログラムの改変)

### (3) その他のデータ構造の種類と適用例(プログラム例)

---

## 第10回 C++の基本(講義+ワークショップ 90分)

---

C++プログラミングの概要として、歴史や特徴について学習する。また、Cと比較した上で、オブジェクト指向プログラミングの利点について学習する。最後に、C++によるプログラムを開発する流れについて習得する。

### (1) C++の説明

1. C++とは(歴史、開発の経緯、適用範囲)
2. C++の特徴(Cと比較した特徴、オブジェクト指向、等)

### (2) オブジェクト指向の説明

1. オブジェクト指向とは
2. オブジェクト指向の特徴(継承、カプセル化、ポリモーフィズム)と利点

### (3) C++による開発の流れ

1. プログラム実行までの流れ(gccによるコンパイル方法)
2. 簡単なC++プログラムの作成

---

## 第 11 回 C++の基本構造(講義+ワークショップ 90 分)

---

C++の基本的な仕組みと構成、基本構文を学習し、プログラミングの内容を理解する。また、サンプルプログラムをもとに、C++で作成したプログラムの流れを確認する。

### (1)C++の書き方の特徴

1. C++のプログラム構成(main 関数、標準ライブラリ、名前空間、等)
2. Cとの相違点(クラス概念、等)

### (2)基本的なプログラム記述の例

1. 変数、定数(変数のスコープ、const 等、C の文法との違い)
2. データ型(bool 型等、C に存在しないデータ型)
3. メモリ管理(new 関数、delete 関数、等)
4. 関数(デフォルト引数、オーバーロード、テンプレート関数、等)
5. 標準ライブラリ(<string>クラス、等)
6. 例外処理
7. サンプルプログラム(標準出力による実行例)

---

## 第 12 回 オブジェクト指向プログラミング(講義+ワークショップ 90 分)

---

C++のクラスの使用方法を学習することで、オブジェクト指向プログラミングの考え方を習得する。また、サンプルプログラムを作成することで、オブジェクト指向プログラミングの有効性を学習する。

### (1)クラスの説明

1. クラスの定義(メンバ、関数の定義、コンストラクタとデストラクタ、スコープ)
2. クラスの包含(ポインタによる包含、参照による包含)

### (2)クラスの継承

1. クラスの継承の定義(オーバーライド、スコープ)
2. 多重継承(多重継承の用途、仮想クラス)
3. サンプルプログラム(クラスの定義、継承の方法)

### (3)オブジェクト指向プログラミングの説明

1. プログラム演習(オブジェクト指向プログラミングによる開発)
2. 構造化プログラムとの比較(C による実装との比較、有効性の確認、等)



---

## 第 13 回 STL(Standard Template Library)(講義+ワークショップ 90 分)

---

コンテナとアルゴリズムのライブラリであり、C++標準の機能である STL(Standard Template Library)の構成や種類について学習する。また、サンプルプログラムをもとに、STL の使用方法について学習する。

### (1) STL の説明

1. STL の構成(コンテナ、非標準コンテナ、イテレータ、アルゴリズム、等)
2. コンテナ(<list>、<map>、<set>、<vector>、等)の種類
3. 非標準のコンテナ(<hash\_map>、<hash\_set>、等)の種類
4. イテレータ(<iterator>)の種類
5. アルゴリズム(検索、ソート、算術演算、等)の種類

### (2) STL によるプログラミング

1. 線形リスト(第 9 回(2)サンプルプログラム)の復習
2. STL による線形リストの実装(<list>)の構成、実装方法、サンプルプログラム)
3. その他データ構造(第 9 回(3))の復習
4. STL によるその他データ構造の実装(データ構造に相当する STL の説明、実装方法、サンプルプログラム)

---

## 第 14 回 GUI アプリケーションの開発(講義+ワークショップ 90 分)

---

ユーザインタフェースを兼ね備えた GUI アプリケーションを開発するために、GUI ライブラリである GTK+と Qt を導入し、使用方法について学習する。

### (1) GTK+による GUI アプリケーション開発

1. GTK+の説明(概要、導入方法、シグナルとコールバックの概念)
2. ウィジェットの種類とパラメータ
3. GDK による描画処理
4. GTK+によるサンプルプログラム(GUI アプリケーション)

### (2) Qt による GUI アプリケーション開発

1. Qt の説明(概要、導入方法、シグナルとスロットの概念)
2. ウィジェットの種類とパラメータ
3. QPainter による描画処理
4. Qt によるサンプルプログラム

---

## 第 15 回 開発ライブラリの使用(講義+ワークショップ 90 分)

---

開発ライブラリを使用する方法を学習し、データベースやコマンドラインオプションの解析など、多機能を備えたアプリケーション開発について習得する。

### (1) 開発ライブラリの説明

1. 開発ライブラリとは
2. 開発ライブラリの種類(データベースライブラリ、コマンドラインオプションの解析ライブラリ、その他ライブラリ)

### (2) データベースライブラリの説明

1. データベースとは(データベースの概念、種類、機能、等)
2. データベースの構成(テーブル、フィールド、等)
3. データベースライブラリの導入(Bereley db ライブラリ、libpq++ライブラリ、等)
4. データベース操作の流れ(ライブラリによる処理の流れ)
5. サンプルプログラム(レコードの取得、更新、等)

### (3) コマンドラインオプションの解析ライブラリの説明

1. コマンドライン解析とは(コマンドライン解析の必要性、等)
2. 解析ライブラリの導入(popt ライブラリ、getopt ライブラリ、等)
3. コマンドライン解析の流れ(ライブラリによる処理の流れ)
4. オプションテーブル(コンテキストの生成、開放、等)
5. サンプルプログラム

以上