1. 科目コード
1242

2. 科目名
プログラミング特論２ (Advanced Programming 2)

3. 担当教員
マルコン　シャンドル (Sandor MARKON)

4. 開講期
秋１期　((昼・夜) 土曜 1-2 時限)

5. 科目の目的･概要
ICT の基礎となるプログラミング技術の実践、特に開発の高信頼性化や効率化をオープンソースソフトウェア(OSS)のソースコードより学ぶ。大規模で実用的なシステム、例えば Linux カーネル等を理解するために必要な背景知識やスキルを「コードリーディング」と名付け、体系的に身に付ける。

6. 科目の学習目標
(1) 実践的な規模のプログラムのコーディング技術・制御構造・データ構造・コーディング規約を理解し、教材システムまたはそれに類似した大規模なシステムを操作・改変できる。
(2) 大規模なプログラムを読むためのコード解析ツールを使えるようになる。
(3) 実践的なシステムの開発プロセスやオープンソース・コミュニティーおよびコードに関する法的規定を理解し、その知識を利用してプロジェクトを実施できる。
(4) 複数の演習問題を解く事を通じてプログラムの解読・デバッギング等、プログラミングの開発スキルを向上する。

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

8. 履修要件

『プログラミング特論①』の学習目標に到達していること。

9. 教科書

なし

10. 参考書

書籍名　：　コードリーディング
著者　：　Diomidis Spinellis
出版社　：　毎日コミュニケーションズ

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	その他
(1)		○				
(2)		○				
(3)					○	
(4)		○				
配点		40			60	

12. 備考

■ 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 オリエンテーション （講義 90 分）

プログラミングの重要性と困難さを例題を通じて分析し、プログラム開発の社会的な位置付けを行い、OSS とコード・リーディングを基本とする学習法を理解する。

1. ソフト開発の概要
   1）実世界におけるプログラミング
   2）プログラミングの問題点
   3）ソフト開発プロジェクトの失敗例
2. 良いプログラムから学ぶ姿勢
   1）OSS を例題として使うメリット
   2）OSS コードの正しい使い方
   3）効率的なコード・リーディング
3. オリエンテーション
   1）授業計画の紹介
   2）ミニプロジェクトの説明とチーム形成

第2回 Python を使ったプログラミング基礎 （講義 90 分）

Python を用いて、後に大規模なシステム構築に必要となるデータ構造、制御構造、プログラム構造を復習し、理解する。

1. Python による会話型プログラミング
   1）Python の起動・文法・操作
   2）インクリメンタルなプログラミング
   3）モジュール・ライブラリー
2. データ構造、プログラム構成
   1）tuple, list, dictionary, 動的データ操作
   2）クラス、オブジェクト、継承、introspection
3. 制御構造
   1）loop, 関数、メソッド
   2）try, except を使った例外処理
   3）コールバック、イーベント駆動処理

第3回 コード・リーディングの基本ツール （講義 90 分）

UNIX（Linux)環境を中心に、ソースコードを効率よく理解・分析するために役に立つツールを学び、実習する。

1. テキスト処理の基礎
   1）正規表現
   2）grep, awk 等の標準ツール
   3）diff 等によるソース操作方法
2. コード・リーディングのツール
   1）ctags, cflow 等の標準ツール
   2）cscope 等、特定用途のツール
   3）エディター等をコード・リーディングに使う方法

第4回 ミニプロジェクトの発足 （フリーディスカッション 90 分）

各チームがミニプロジェクトのテーマを宣言する。またこれまでの講義を振り替えて、理解を深める。

1. ミニプロジェクト発足
   1）各チームのテーマ・計画宣言
   2）コメント・ディスカッション
2. C プログラミングの基礎を復習

第5回　問題事例の解析　（講義 90 分）

実可動したシステムで起きた問題について、これまで学んだツールを使って解析し、討議する。

1. 例題プログラムの解読
2. 動作の解析
3. 問題点の理解
4. 対策、発生防止について検討・討議

第6回　中級 C プログラミング　（講義 90 分）

実用的なプログラムで必要となるデータ構造や制御構造について学ぶ。

1. データ構造
   1）各種配列
   2）リスト・スタック・ハッシュ
2. 制御構造
   1）プロセス・シグナル
   2）スレッド・セマフォアー

第7回　at(1)と cron(1)のコード・リーディング　（講義・実習 90 分）

BSD のコマンドを読んで、一段と上位な C プログラミングを学ぶ。

1. at(1)のコード・リーディング
   1）man at 等で仕様を理解する
   2）静的構造を分析する
   3）実験しながら動作を理解する
2. cron(1)のコード・リーディング（宿題）

第8回　組み込みシステムのプログラミング　（講義 90 分）

家電製品から発電所までの各種システムのプログラミングの基礎について理解する。

1. 組み込みシステムの概念と特徴
2. 組み込みシステムのハードとソフト
3. 実時間システムの条件と実現法
4. クロス開発・デバグやテストの手法

第9回　ミニプロジェクト中間発表　（フリーディスカッション 90 分）

各チームがミニプロジェクトの進捗状況を発表し、討議する。

1. ミニプロジェクト発表
2. 中級 C プログラミングの復習

第10回　プログラミングの規程・規約・習慣　（講義 90 分）

プログラミング言語のルールを超える、特にチームワークで必要となる様々な決め事を習う。

1. ソースコードの書き方
   1)ファイルやディレクトリの構造
   2)ソースコードの構造
2. プロジェクトマネジメント
   1)プロジェクトの構成
   2)開発プロセスの管理

第11回 Apache のコード・リーディング(1) (講義・実習 90 分)

学習したツールを使って、Apache のソースを読んで、大成功を納めた名品から知識を得る。

1. コード・リーディングのロードマップ
2. Apache を読む

第12回 C 以外のプログラミング言語の紹介 (講義 90 分)

プログラミングをより効率的に行うため、各タスクに合った言語を選べる様に、幾つかの言語について習う。

1. bash, awk を使った日常的なプログラミング
2. 高信頼性の Erlang プログラミング
3. 高性能の Common Lisp プログラミング
4. プログラミング言語選定(Java,PHP,Perl,...)

第13回 Apache のコード・リーディング(2) (実習 90 分)

Apache のコード・リーディングを継続し、知識を深める。

第14回 ミニプロジェクトの成果発表 (フリーディスカッション 90 分)

ミニプロジェクトの成果や経過・解決した問題等を報告し、全員で討議する。

1. 各チームの発表
2. 討議

第15回 ミニプロジェクトの評価と反省 (フリーディスカッション 90 分)

前回の発表やレポートを基に、プロジェクトを評価する。またコース全体について反省し、今後の発展を討議する。

1. 評価発表
2. 反省会