

■科目基本情報

|             |  |                                  |    |       |      |       |    |
|-------------|--|----------------------------------|----|-------|------|-------|----|
| 科目名         | プログラミング演習  |                                  |    | 科目コード | J109 | 科目区分  | 専門 |
| 学科・コース      | 情報システム工学科 メディアデザインコース  |                                  |    | 学 年   | 1 年  | 学 期   | 通年 |
| 方 式         | 演習   | 必 選                              | 必修 | 単 位   | 3    | 総 時 数 | 90 |
| 実務経験のある教員科目 | <input checked="" type="checkbox"/> 対象・ <input type="checkbox"/> 対象外 | ソフトウェア開発企業にて業務アプリケーションの設計、開発に携わる |    |       |      |       |    |

■授業詳細情報

|           |  |  |  |  |  |  |  |
|-----------|--|--|--|--|--|--|--|
| 授 業 概 要   | プログラミングを行うために必要な開発環境構築実習を実施する。またその仕組みを理解するために講義する。プログラミング言語 Java を使用し、データ型や変数、プログラミングの文法及び基本的な命令の利用方法を理解するため、講義を交え演習を行う。学習した基本技術をもって簡単なプログラムから開始し、進度に応じて複雑なプログラミング演習を行う。 |  |  |  |  |  |  |
| 達 成 目 標   | 基本的な開発環境を構築することができる。Java 言語の言語仕様を十分に理解し、変数や各種命令の利用及び、適切なクラス、メソッドの定義ができる。また、オブジェクト指向開発を理解し、それらをもとに簡単なアプリケーションを作成することができる。   |  |  |  |  |  |  |
| 使 用 教 材   | 教科書： スッキリわかる Java 入門 第4版<br>副教材： 演習問題：Google Classroom にて出題及び提出、確認を行う<br>参考書：  |  |  |  |  |  |  |
| 授 業 外 学 習 | 出題された演習問題は期限までに提出すること。理解度を深めるためにも授業対象箇所の予習をしておくこと。   |  |  |  |  |  |  |

授業計画

| 項 目    | 内 容  | 時数 |
|--------|--|----|
| 1. 第1部 | <ul style="list-style-type: none"> <li>●プログラムの書き方                             <ul style="list-style-type: none"> <li>・ Java プログラム開発の基礎知識</li> <li>・ Java プログラムの基本構造</li> <li>・ 変数宣言</li> </ul> </li> </ul>   | 2  |
|        | <ul style="list-style-type: none"> <li>●式と演算子                             <ul style="list-style-type: none"> <li>・ 計算の文</li> <li>・ オペランド</li> <li>・ 評価の仕組み</li> <li>・ 演算子</li> <li>・ 型の変換</li> <li>・ 命令実行の文</li> </ul> </li> </ul>                                 | 4  |
|        | <ul style="list-style-type: none"> <li>●条件分岐と繰り返し                             <ul style="list-style-type: none"> <li>・ プログラムの流れ</li> <li>・ ブロックの書き方</li> <li>・ 条件式の書き方</li> <li>・ 分岐構文のバリエーション</li> <li>・ 繰り返し構文のバリエーション</li> <li>・ 制御欧文の応用</li> </ul> </li> </ul> | 8  |
|        | <ul style="list-style-type: none"> <li>●配列                             <ul style="list-style-type: none"> <li>・ 配列のメリット</li> <li>・ 配列の書き方</li> <li>・ 配列と例外</li> <li>・ 配列のデータをまとめて扱う</li> <li>・ 配列の舞台裏</li> <li>・ 配列の後片付け</li> <li>・ 多次元配列</li> </ul> </li> </ul>   | 10 |
|        | <ul style="list-style-type: none"> <li>●メソッド                             <ul style="list-style-type: none"> <li>・ メソッドとは</li> <li>・ 引数の利用</li> <li>・ 戻り値の利用</li> <li>・ オーバーロードの利用</li> <li>・ 引数や戻り値に配列を用いる</li> <li>・ コマンドライン引数</li> </ul> </li> </ul>             | 4  |
|        | <ul style="list-style-type: none"> <li>●複数クラスを用いた開発                             <ul style="list-style-type: none"> <li>・ ソースファイルを分割する</li> </ul> </li> </ul>   | 8  |

|               |   |  |
|---------------|---|--|
| <p>2. 第2部</p> | <ul style="list-style-type: none"> <li>・複数クラスで構成されるプログラム</li> <li>・パッケージを利用する</li> <li>・パッケージに属したクラスの実行方法</li> <li>・名前空間</li> <li>・JavaAPI について学ぶ</li> <li>●オブジェクト指向             <ul style="list-style-type: none"> <li>・オブジェクト指向を学ぶ理由</li> <li>・オブジェクト指向の定義と効果</li> <li>・オブジェクト指向の全体像と本質</li> <li>・オブジェクトと責務</li> <li>・オブジェクト指向の3大機能と今後の学習</li> </ul> </li> <li>●インスタンスとクラス             <ul style="list-style-type: none"> <li>・仮想世界の作り方</li> <li>・クラスの定義方法</li> <li>・クラス定義による効果</li> <li>・インスタンスの利用方法</li> </ul> </li> <li>●様々なクラス機構             <ul style="list-style-type: none"> <li>・クラス型と参照</li> <li>・コンストラクタ</li> </ul> </li> <li>●継承             <ul style="list-style-type: none"> <li>・継承の基礎</li> <li>・インスタンスの姿</li> <li>・継承とコンストラクタ</li> <li>・正しい継承、間違った継承</li> </ul> </li> <li>●高度な継承             <ul style="list-style-type: none"> <li>・未来に備えるための継承</li> <li>・高度な継承に関する2つの不都合</li> <li>・抽象クラス</li> <li>・インターフェース</li> </ul> </li> <li>●多態性             <ul style="list-style-type: none"> <li>・多態性とは</li> <li>・ざっくり捉える方法</li> <li>・ざっくり捉えたものに命令を送る</li> <li>・捉え方を変更する方法</li> <li>・多態性のメリット</li> </ul> </li> <li>●カプセル化             <ul style="list-style-type: none"> <li>・カプセル化の目的とメリット</li> <li>・メンバに対するアクセス制御</li> <li>・getter と setter</li> <li>・クラスに対するアクセス制御</li> <li>・カプセル化を支えている考え方</li> </ul> </li> </ul> | <p>6</p> <p>6</p> <p>6</p> <p>6</p> <p>6</p> <p>6</p> <p>6</p> <p>12</p> |
| <p>3. 第3部</p> | <ul style="list-style-type: none"> <li>●演習</li> </ul>   | <p>12</p>  |
| <p>評価方法</p>   | <p>①期末試験（又は各検定試験への合格による評価）：60%、②出席率：20%、③課題提出：20%<br/>                 ①～③の合計得点を評価（優、良、可、不可）に置き換える<br/>                 ※Java プログラミング能力認定試験へ合格した場合、定期試験を合格としこれを免除する。</p>  |  |
| <p>関連科目</p>   | <p>1年次：アルゴリズムとプログラミング、情報テクノロジー<br/>                 2年次：組み込みシステム概論、ハードウェア技術、情報理論、情報システム設計と開発、ソフトウェア開発と演習</p>  |  |
| <p>備考</p>     | <p>Java 言語は基幹業務システム開発などで主要言語として利用されています。また、オブジェクト指向言語としても代表的なものとなっており、プログラミングそのものを理解するためにも最適です。しかし、習得には時間がかかりますので、理解を深めるためにも予習や復習、また積極的に質問をし、問題を解決してください。</p>   |  |