

# 産学連携実績紹介フォーム

## 1. 講座の計画から実施までの情報

教育機関名 (学校名・学部学科等)	東洋大学 総合情報学部 総合情報学科	実施時期	2014年度(前期)  (5年目:赤字は改善点)
対象学年・学期・人数	2年次 約100名 (50名のコースを2コース開講)		
講座名	「実システムのプログラミング基礎」(登録科目名:プログラミング技法Ⅱ)		
連携企業・団体	株式会社 富士通ラーニングメディア		
支援・連携の類型	企業より提供された教材・事例を使用して、講義・演習は大学教員が実施		
講座の概要・特徴	本講座は、コーディングに関するプログラミング科目の最終段階である。プログラミング系科目には、初心者を対象に一般的な構造型言語による原理を学ぶ「入門プログラミング」、Javaに基づくオブジェクト指向プログラミングを学ぶ「プログラミング技法I」がある。本「プログラミング技法II」はオブジェクト指向プログラミング技法を実際のアプリケーションに適用することを目指す。本講座の後継としてUMLに基づくソフトウェア設計を行う「ソフトウェア工学I/II」がある。		
産学連携検討の背景	総合情報学部は1999年に新設された文理融合を理念とする情報系学部である。このような学部では理系のみならず文系の学生へもプログラミングスキルのすそ野を広げる必要がある。しかし、質を薄めるのではなく、一定の濃さを保ったまま、量を広げることは容易ではない。そのためには学生の品質保証を兼ねて企業の実務に準じた基準で検証する必要がある。そこで、本講座では企業の研修プログラムに基づくコンテンツを採用することで上記目的を達することとした。		
連携の狙い、目的・目標	<p>産業界では、入社後すぐに活躍できる“即戦力”を求める声が強まっているが、即戦力の中核となるのは、この講座が対象とする実践的なプログラミング技術である。</p> <p>この講座は、基本的なプログラミングスキルを習得した学生が、即戦力として大きくステップアップするための講座として位置づけられる。授業では、企業で実際に用いられているソースコードを用いて、コードを読む訓練と書く訓練を繰り返し行う。学生に対して敢えて手加減せず、企業が提供する難易度の高い素材を用いることで、企業における実務の水準を学生に体験させる。また、作法や規約の重要性など、実務の基本となるソフトウェア工学の基礎についても理解が深まる。企業の実務の一部を体験できるこの講座の受講によって、学生は、学部の早い段階から、IT業界における実際の仕事のイメージをつかむことができるようになる。</p> <p>また、本講座は、プログラミング演習の総仕上げであり、ソフトウェア工学を学ぶための前提科目として位置づけている。授業では、Java のライブラリを活用したグラフィックス、イベント処理、並行処理、GUI について学び、演習を通じて、企業で実際に使用されているソースコードを利用したやや難易度の高い課題に取り組む実践性の高いカリキュラムとなっている。</p> <p>演習の中では、コードリーディングおよびコードライティングの両方を実施し、システムエンジニアとして他人が書いたコードを読解する能力を高めるとともに、コードライティングの練習を重ねることで、100行程度のプログラムを自力で記述できる力を養成する。</p>		
連携にあたっての課題・懸念	本来少人数を対象とした研修プログラムを中～大人数に適用すること、学部卒業者を対象とした内容を学部2年に適用することなどが懸念された。それぞれに対して企業コンテンツを調整することで運用可能とした。2014年度にはソースの公開が可能のように、コードを全面的に書き換えた。		
講座の位置づけ 既存講座との関係	本講座は、コーディングに関するプログラミング科目の最終段階である。プログラミング系科目には、初心者を対象に一般的な構造型言語による原理を学ぶ「入門プログラミング」、Javaに基づくオブジェクト指向プログラミングを学ぶ「プログラミング技法I」がある。本「プログラミング技法II」はオブジェクト指向プログラミング技法を実際のアプリケーションに適用することを目指す。本講座の後継としてUMLに基づくソフトウェア設計を行う「ソフトウェア工学I/II」がある。		
履修前提条件	<p>&lt;前提履修科目&gt;</p> <ul style="list-style-type: none"> <li>・プログラミング技法 I</li> <li>・Java 言語の文法に関する知識を有すること</li> <li>・オブジェクト指向に関する知識があること</li> </ul>		

授業準備と実施の体制	<ul style="list-style-type: none"> <li>・2013年6月7日(金)～7月26日(金) 毎週<b>金曜日</b>の1コマ (計15コマ)</li> <li>・主担当: 上原 稔(総合情報学部総合情報学科 教授) 他1名</li> <li>・TAとして他学部学生をコース毎に1名配置</li> </ul>
成績評価の方法	<p>&lt;知識・技術&gt;</p> <ul style="list-style-type: none"> <li>・試験60% 段階的に小テストを実施し、知識及び技能を評価する。</li> <li>・レポート40% 総合演習により到達度を検証する。</li> </ul>

講座の構成(シラバス)	単元と時間配分 (1コマ90分で実施)	演習・実習	実施担当・役割分担
	(1) ガイダンスおよび序論 <ul style="list-style-type: none"> <li>・科目ガイダンス</li> <li>・プログラミング技法 I の復習</li> <li>・Javaを用いたアプリケーション例</li> </ul>	講義 + 演習	全て大学教員が実施
	(2) グラフィックス <ul style="list-style-type: none"> <li>・アプレット</li> <li>・グラフィックス</li> </ul>	講義 + 演習	全て大学教員が実施
	(3) イベントとイメージ <ul style="list-style-type: none"> <li>・イベント駆動の処理方式</li> <li>・イメージを用いたプログラムの作成</li> </ul>	講義 + 演習	全て大学教員が実施
	(4) アプレットの応用事例 <ul style="list-style-type: none"> <li>・ゲームを題材としたアプレットの事例紹介</li> </ul>	講義 + 演習	全て大学教員が実施
	(5) AWTレイアウト <ul style="list-style-type: none"> <li>・AWTの概要</li> <li>・レイアウトマネージャ</li> </ul>	講義 + 演習	全て大学教員が実施
	(6) AWTコンポーネント <ul style="list-style-type: none"> <li>・コンポーネントの使い方</li> <li>・パネルによる組み合わせ方</li> </ul>	講義 + 演習	全て大学教員が実施
	(7) Windowアプリケーション <ul style="list-style-type: none"> <li>・フレームのあるアプリケーションの作成</li> <li>・ダイアログの使い方</li> </ul>	講義 + 演習	全て大学教員が実施
	(8) ファイル操作 <ul style="list-style-type: none"> <li>・データモデル</li> <li>・テキストファイルおよびバイナリファイルの読み書き</li> </ul>	講義 + 演習	全て大学教員が実施
	(9) 並行プログラミング <ul style="list-style-type: none"> <li>・スレッドプログラミング</li> <li>・排他制御</li> <li>・デッドロック</li> <li>・同期</li> </ul>	講義 + 演習	全て大学教員が実施
	(10) アニメーション <ul style="list-style-type: none"> <li>・ダブルバッファリング</li> </ul>	講義 + 演習	全て大学教員が実施
	(11) リアルタイムAWTアプリケーションの事例 <ul style="list-style-type: none"> <li>・リアルタイムゲーム</li> </ul>	講義 + 演習	全て大学教員が実施
	(12) ネットワーク <ul style="list-style-type: none"> <li>・ソケット</li> <li>・クライアント・サーバ</li> </ul>	講義 + 演習	全て大学教員が実施
	(13) ビジネスアプリ <ul style="list-style-type: none"> <li>・簡単なオンラインショップの作成</li> </ul>	講義 + 演習	全て大学教員が実施
	(14) 総合演習(1) <ul style="list-style-type: none"> <li>・自由課題</li> </ul>	講義 + 演習	全て大学教員が実施
	(15) 総合演習(2) <ul style="list-style-type: none"> <li>・課題の検証</li> </ul>	講義 + 演習	全て大学教員が実施

講座ならびに演習・実  
習の具体的な進め方

・PC教室 (Windows、Java、Eclipse) による演習環境の整備

## 2. 講座実施後の情報

受講者の声(受講目的、修得目標)	(2014年度) 本学授業評価アンケート実施の結果、「授業のわかりやすさ」については、教材の変更に伴い十分な資料がそろっていない点が課題となった。また、「授業運営について」は概ね良好であったが、「授業に集中する環境作り」に課題が残った。「学習成果について」は「新しいものの見方や考え方が身に付いた」という一方で課題の難易度が上がり学生の熱意が下がる傾向があった。問題点は教材の難易度と資料の不足に集約される。
受講者の感想(本講座で得られたもの)	(2014年度) 予習・復習の資料がもっと欲しい。 進み方が早すぎる。 プログラムを読むだけでなく、書く課題をもっとやりたかった。
先生の評価	(2014年度) 2014年度より、実践的コードリーディングを目指し、本コースの内容を抜本的に変更した。すべての教材を独自に準備し、オープンソース化を進めた。APIを理解するために短い基本コードで試しながら学べるようにした。同時に、コードの読解力を養うために実際のアプリケーションに近い100行程度の長い応用コードを用意した。応用コードは実用的な意味を持たせながら、可能な限り簡潔に記述した。 前提科目の中には概念的な課題が多かったため理解が困難であった学生も、目で見て動作がわかる例題には関心が高かった。しかし、昨年より教材が高度化し、含まれる概念も多様化している。そのため、消化不良となってしまった。これは難易度の適正化は今後の課題である。
今後の展望(継続に向けた課題)	(2014年度) 実用性の高いアプリケーションに関する関心は高く、教材の変更はある程度成功している。しかし、難易度が上昇してしまったため、理解が不十分となったことは問題である。また、予習・復習の教材が間に合わず不十分であった点も問題である。次年度は、基本コードと応用コードに二極化させ、応用コードの読解に十分な時間を用意する。また、コードに適切なコメントを付加し、理解を促す。

学期末にコンピテンシーに関する学生の自己評価を行った。

### 1.評価項目と評価区分（レベル記述： 0問題行動 1受動的行動 2能動的行動 3卓越行動）

#### Q1 読解力 教科書が理解できる

- 0 教科書を持っていない
- 1 指示された箇所だけ読む
- 2 指示されなくても読む
- 3 事前に教科書を読んで授業に望む

#### Q4 倫理

- 0 宿題をコピーする
- 1 宿題の解き方を教えてもらう
- 2 宿題の解き方を教えてあげる
- 3 宿題以外も教えてあげる

#### Q2 計画実行

- 0 何をしたらよいのかわからない
- 1 指示された宿題がわからなければあきらめる
- 2 指示された宿題がわからなければ質問する
- 3 予習し、わからなければ自分で調べる

#### Q5 自己管理

- 0 しばしば遅刻する
- 1 しばしば宿題を忘れる
- 2 体調に気をつけ遅刻はしない。宿題は出来るだけ提出する。
- 3 授業時間内に解答できるよう、あらかじめ予習しておく

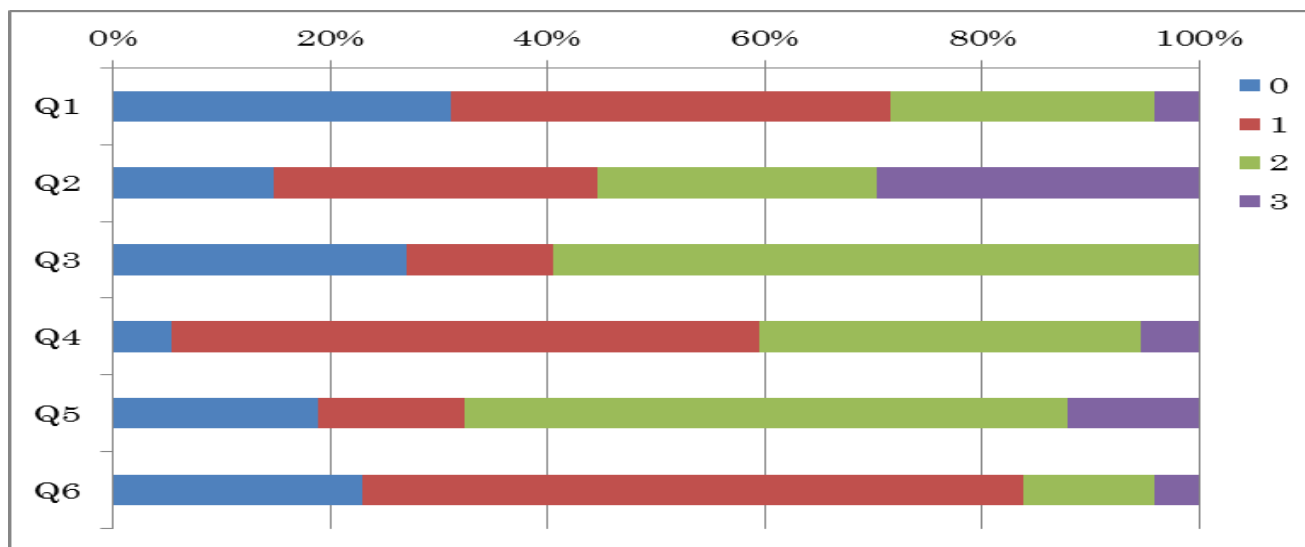
#### Q3 知識獲得力

- 0 授業だけでは理解できない
- 1 宿題をするとわかる
- 2 わからない箇所は自分で調べる
- 3 授業だけでよくわかる

#### Q6 専門知識

- 0 授業が理解できない
- 1 個々の技術は理解できるが、組合わせた問題は自力で解けない
- 2 組み合わせた問題も自力で解ける
- 3 総合問題を発展させ、改良できる

### 2.評価区分の分布



### 3.評価結果

Q1	読解力	問題行動が多いように見えるが、教科書を使用していないことを「教科書を持っていない」ととらえてしまったためである。特定の教科書を指定せず、教材を配布しているので、この自己評価は当たらない。
Q2	計画実行	卓越行動が非常に多かった。これは予習を強制したためである。予習の実施率は極めて高かった。
Q3	知識獲得力	能動的行動が非常に多く、多くの学生が自主的に知識を獲得する行動を選択していた。一方で、「授業だけでは理解できない」とするものも多かった。これは設問の表現が不適切であることも理由の一つであろう。「授業だけでは理解できない」は本来の「授業以外の学習をしなかった」という意味が通じていなかった。
Q4	倫理	問題行動は非常に少なかった。
Q5	自己管理	能動的行動が非常に多かった。一方で、問題行動を継続し改善されない者がいた。
Q6	専門知識	受動的行動が非常に多かった。難易度が不適切であったためと考えられる。

### 4.総評

今回、コンピテンシー評価を実験的に実施した。

すべてのコースで徹底できなかつたため、学期の前後で比較できなかつた。

また、設問に不適切な選択肢が含まれていたなど、様々な問題点の洗い出しができた。

コンピテンシー評価は授業内容と密接に関連し、授業改善と同時に実施する必要がある。

## 3. 講座開設時の支援企業・団体からの情報

提供教材・コンテンツ情報	新入社員向け「Java プログラミング (AWT 編)」		
提供元	株式会社 富士通ラーニングメディア	費用(標準価格)	応相談
支援の目的・目標	<p>Java アプレットおよびAWT (Abstract Window Toolkit)について、実習をとおして学習する。</p> <p>1. アプレットについて説明できる。 2. AWT について説明できる。</p>		
具体的な支援内容または提供教材の内容	<p>提供教材の構成</p> <p>1 アプレットとは 2 アプレットの作成手順 3 アプレットクラスの定義 4APPLET タグ 5 アプレットのライフサイクル 6 ライフサイクルに関するメソッド 7 グラフィックスの描画 8 色の設定 9 フォントの設定 10 グラフィックスの再描画 11 イメージの表示 12HTML パラメータの読み込み [実習問題1-1] [実習問題1-2] [実習問題1-3] 13AWT とは 14 コンポーネント 15 コンテナ 16 コンテナとコンポーネント 17 レイアウトマネージャ 18AWT コンポーネント 19AWT イベントモデル 20 イベント処理の定義 21Java Window アプリケーション [実習問題2-1] [実習問題2-2] [実習問題2-3]</p>		
講座実施における企業・団体の役割	<p>・新入社員研修で実際に使用している教材・ソースコード、実習問題の課題提供 ・総合演習の応援講師派遣</p>		
企業・団体からの推薦コメント	<p>本研修を受講する新入社員の中でも、学生時代の学習したプログラムと実務で求められるプログラムの難易度の差に驚いてしまう方もいます。そのようなことにならないよう、早い段階で実務で求められるレベルに触れることが大切です。本研修では、例題プログラムや実習問題の作成を通じて、自分自身の現状のスキルレベルを把握することが可能です。本研修を通じて、その後の大学生活における学習のモチベーションへとつながることができれば幸いです。</p>		