

# 第7章 サイバー大学における演習講義の取り組み： JAVA プログラミング演習事例報告

横山輝明<sup>1</sup>

## 要旨

本稿では、著者が担当する Java プログラミング演習における演習型講義の運用について報告する。通常の講義と異なり、受講生側での課題の実施が必要となる演習では、受講生が円滑に課題を実施するための支援が必要となる。特に、プログラミングではプログラム言語の知識という形式知に限らず、プログラムを作成する際のコツという暗黙知の学習が重要である。Java プログラミング演習においても、教科書の設定、スクリーンキャストの提供、TA の活用などの工夫を施している。これらの工夫によって、オンライン型演習において、形式知や暗黙知の伝授、そして例外的な個別対応までを可能とする運用体制を実現している。

キーワード：オンライン型演習、プログラミング演習、スクリーンキャスト

## 1. サイバー大学における演習授業

サイバー大学は 2007 年度に設立された株式会社立の大学である [Cyber-U]。サイバー大学は完全インターネット大学として設置されており、全ての授業は WWW サイト上でのビデオ配信として提供される。ビデオでは、教員からの授業説明の動画と授業内容に基づくスライド資料を配信する。図 1 が実際の授業の様子である。学生はこのビデオ配信の視聴によって受講する。また、受講生や教員の間でのコミュニケーションのために各授業に掲示板が設置されている。教員への質問や受講生間での議論にはこの掲示板が利用される。



図 1 サイバー大学の授業配信システム

1 サイバー大学 IT 総合学部 講師

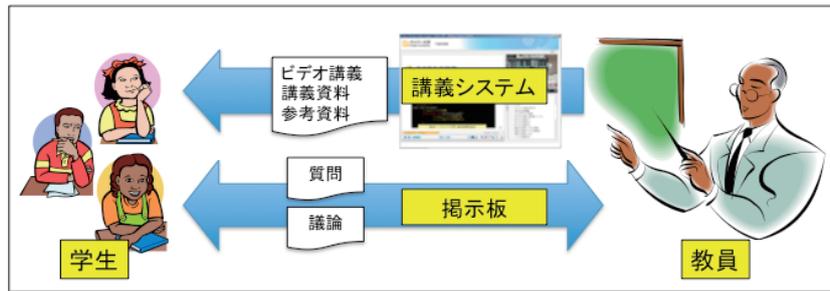


図2 サイバー大学における授業提供形態

サイバー大学が提供する授業には、講義と演習の二種類がある。講義は知識伝達を目的とする従来の座学型式の授業である。講義ではビデオとスライドを用いて授業内容を伝える。講義の目的は授業で説明される知識の習得となる。一方で、習得した知識の実践を図るのが演習である。演習では、受講生は習得した知識を用いてさまざまな課題に取り組む。課題の実施を通じて、習得した知識の確認や実践的な知識活用が演習の目的となる。演習の実施では、受講生の課題遂行を適切に支援する必要がある。学生に要求される課題内容は演習科目ごとに異なるため、それぞれの演習にて内容に応じた工夫が取り組まれている。

本稿では、著者が教員として提供している「Java プログラミング演習」を取り上げて、演習授業の実施における工夫について報告する。プログラミング学習では、ただプログラムの知識をつけるだけでは不十分で、実際にプログラムを作るための考え方を理解することが重要である。プログラミングの実践的知識を与えるための、プログラム作成途中の失敗や試行錯誤などのプログラミング過程も含めた演習について報告する。

## 2. Java プログラミング演習

### 2.1. 演習授業の概要

サイバー大学の演習は、基本的には講義と同じシステムを用いて提供される。その運用ルールも講義と同様である。多くの演習では前提となる知識を伝える講義が提供されており、学生にはそれらの講義は同時受講を薦めている。

演習も講義と同じく全15回の授業として提供される。演習の授業は、演習内容や課題出題などの説明パートと演習課題の実施パートから構成される。演習の説明パートでは課題の実施に必要な情報が提供される。この説明は通常の講義と同様にビデオとスライドを用いる。演習では、説明パートが主体ではないためこのビデオの収録は教員各自での自主的な収録が認められている。課題の実施パートでは、受講生自身が課題に取り組む。内容説明パートと課題実施パートを合わせて学習時間としている。

### 2.2. Java プログラミング演習の運用

著者が担当する、プログラム言語 Java を学習する演習形式の授業が Java プログラミン

表1 Javaプログラミング演習の講義内容

1回	Java 概要	9回	アルゴリズム
2回	Java 環境の構築	10回	プログラム実例：グラフィックス
3回	Java プログラムの基本	11回	プログラム実例：HTTP 通信
4回	基本要素	12回	プログラム改造
5回	制御構造	13回	最終課題出題
6回	オブジェクトとクラス定義	14回	最終課題の進捗報告
7回	入出力	15回	まとめ
8回	これまでのおさらい		

表2 Javaプログラミング演習の科目目標

<p><b>1. 科目概要</b></p> <p>Javaはネットワーク関連の機能を標準で備える、オブジェクト指向プログラミング言語である。そのため、これまでの既存の言語に比べて学習が容易でインターネットとの親和性が高いという特徴を持ち、適用領域はますます拡大している。本演習では、履修者にJava言語の基本的な概念を習得し、ネットワークプログラムを作成することが可能になることを目標とする。</p> <p><b>2. 科目目標</b></p> <p>本演習では、履修者が初等的なJava言語のプログラミング経験を得ることを目的とする。特にJava言語に触れたことのない初級者を対象として、簡単なJavaプログラムの作成やJavaプログラムの改造などを行えるようになることを目標にする。</p> <p><b>3. 履修前提条件</b></p> <p>演習ではパソコンのコマンドライン上でJavaコンパイラを利用するため、コマンドライン上の操作を習得していることが望ましい。(出来る限りのサポートはします。)</p> <p>演習で利用する教科書は各自で購入してください。</p>
--

グ演習である。プログラム初級者を対象として、プログラム言語Javaを用いた実習を提供する。表1はJavaプログラミング演習の全15回の授業内容である。演習では、Javaのインストールから、簡単な文法学習、制御構造の理解、グラフィックスプログラムや通信プログラムの事例学習と改造を題材として学習する。表2はJavaプログラミング演習の科目目標などを記したシラバスである。演習を通じて、Java言語の文法やアルゴリズムの考え方を理解して、簡単なプログラムを自分で作ることができるようになることが目標である。

プログラミングへの理解の確認のために、講義の途中ではレポート課題を出題している。本演習ではプログラミング学習を、①アルゴリズムの考え方の理解と、②プログラム作成の二段階に分けている。そこで、①の確認として、Java言語の基本的な文法や制御構造を習得した後に、アルゴリズムへの理解を確認するためにプログラム作成の課題を出題している。

レポートや最終課題にて、作成したプログラムを成果物として提出してもらうことで演習授業での学習目標が達成できたか確認をしている。学生の理解や前提知識に差があるた

表3 Javaプログラミング演習の最終課題

最終課題：以下のいずれかを選んでプログラムを作成して提出する (1) グラフィックプログラムの改造 (2) 通信プログラムの改造 (3) プログラムの自作
--

表4 Javaプログラミング演習の評価配分

視 聴	平常点	レポート	小テスト	期末試験	合 計
0%	0%	20%	30%	50%	100%

め、最終課題でのプログラム作成は選択式として理解度の高い学生にはより難易度の高いプログラム作成に挑戦できるように設定している。具体的には表3のとおり、演習にて提示したサンプルのプログラムを改造することから、自分でテーマ設定することまでを許している。また、学生にとって継続してプログラム学習に取り組んでもらえるように、最終課題に対してはさらに次の目標となるようなプログラム改良などのコメントを与えて演習授業を終了している。表4はJavaプログラミング演習における成績評価の配分である。各回の理解を確認する小テストと成果物を合わせて成績評価としている。

### 2.3. 演習の参考資料

Javaプログラミング演習では、Java言語の説明や演習課題を説明するために教員からの説明ビデオとスライド資料を提供している。スライド資料はpdfファイルとして、受講生が保存可能な形でも提供している。また、その他の学習資料としてJava言語に関する書籍を教科書として指定している [BookJava]。さらに、プログラムの手本となるサンプルソースコードや、プログラム作成の途中過程を収録したスクリーンキャストと呼ばれる動画も配布している。

### 2.4. 演習授業の収録

Javaプログラミング演習のビデオは、著者が研究員として所属する、奈良先端科学技術大学院大学にて収録している。図3が収録機材である。iPhone 3GSを三脚で固定して、



図3 iPhoneを用いた収録機材

イヤホンマイクを接続して利用している。演習のビデオ部分はこの機材で十分な収録が可能である。これまでの学生からのフィードバックより、教員説明では音声品質が重要との知見を得ている。そこで、静かな深夜の時間帯を選んでの収録と、携帯電話での通話用イヤホンマイクによって、環境ノイズの少ない音声収録になるように心がけている。収録した動画は、不要部分の編集の後に、サイバー大学の講義システムでの配信形式である Windows Media Video 形式に変換して大学側に提出している。そのファイルが大学システムより授業として配信される。

## 2.5. 演習の開講状況

開学より5年間、本講義を提供してきた。プログラミング入門としての側面もあるので、主に低学年の学生が受講する講義となっている。

## 3. Java プログラミング演習での問題

Java プログラミング演習では、いくつかの演習実施上の問題がある。ここでは、それらの問題のうち代表的なもの3つを説明する。

### 3.1. Java 言語についての講義

本学の演習講義は、基本的には授業講義とペアとして提供されている。たとえば、「ネットワークプログラミング演習」という演習講義には「ネットワークプログラミング論」という授業講義が提供されている。ところが、Java プログラミング演習ではカリキュラム編成の都合上、授業講義が用意されていない。

プログラミングの理解には、プログラム言語を構成する命令や文法などの基礎知識の習得がまず必要となる。受講生にそれらの基礎知識を伝えるために、限られた授業時間のなかで前提知識の講義と演習の実施を両立させる授業設計が求められる。

### 3.2. プログラミング過程の支援

Java プログラミング演習では、受講生にプログラミングを実施する。プログラミングでは、プログラムのエラーやその訂正、また段階的な実装など、その過程においてさまざまな問題が発生する。プログラミングの学習では、これらの問題についての試行錯誤や解決までの考え方を理解してもらうことが重要である。

そこで、各受講生に対してプログラミング過程における支援が必要となる。対面型授業であれば、それぞれの学生の様子を直接的に把握して個別に指導することが容易であるが、遠隔地から非同期に演習を実施するオンライン型授業では演習の支援にはなんらかの工夫が必要となる。

### 3.3. プログラミングレベルの差

社会人学生が多く受講しているサイバー大学では、受講生のもつプログラミングへの知識にも大きな差がある。すでに業務上でプログラミングを経験している者から、Java 言語を通じてプログラミングを初めて体験するものまで、さまざまな受講生が存在する。

初級者のみに注目しすぎると、中級以上の受講生にとって授業内容が薄くなってしまふ。しかし、演習で取り組む全てが初めてのこととなる初級者を放置しては授業の目標となるプログラミングへの理解は達成できない。そのため、初級者の理解も促しつつ、中級者以上の受講生にも得るものがあるような内容設定が重要である。

## 4. 演習授業での工夫

Java プログラミング演習を円滑に実施するために、いくつかの工夫を施している。ここでは、それらの工夫について説明する。特に、教員のプログラミング過程をお手本として配布するスクリーンキャストや、個別対応を可能とするための TA の活用などはプログラミングの過程を支援するための独自の工夫である。

### 4.1. 教科書の採用

プログラミング入門という講義の位置づけを留意して、「改訂 新 Java 言語入門（ビギナー編）（著 林晴比古）」[BookJava] を採用している。本書は Java 言語を知らない入門者を対象としており、講義内容も本書に合わせて設定した。講義中でも、講義内容について教科書との対応関係を言及することで、学生の自主的な学習を促している。また、講義パートに利用可能な時間は限られているため、教科書を利用することで客観的な事実や知識の伝達は各学生の望むペースで進めることが可能となっている。

### 4.2. スクリーンキャストの利用

スクリーンキャストとは、PC の画面を録画したものである [SCast]。スクリーンキャストを作成するために、さまざまな OS 上や記録形式に対応したツールが登場している [Wink] [WMEnc]。スクリーンキャストは実際の PC 操作の様子を伝えるため、各種 PC 操作の説明などにおいて広く利用されている。本演習でも、Java プログラミングの作成途中の考え方を伝えるためにスクリーンキャストを利用している。

プログラミング学習では、その文法や命令など知識面での習得とそれらの知識を用いてプログラム作成の考え方の習得が必要になる。これは、英語学習における英単語学習と英作文作成の関係に似ている。プログラミングに必要な知識については客観的に提示することができるが、プログラム作成の考え方については提示することが困難であった。スクリーンキャストを利用することで、試行錯誤も含めたプログラム作成途中の様子を伝えることに取り組んでいる。

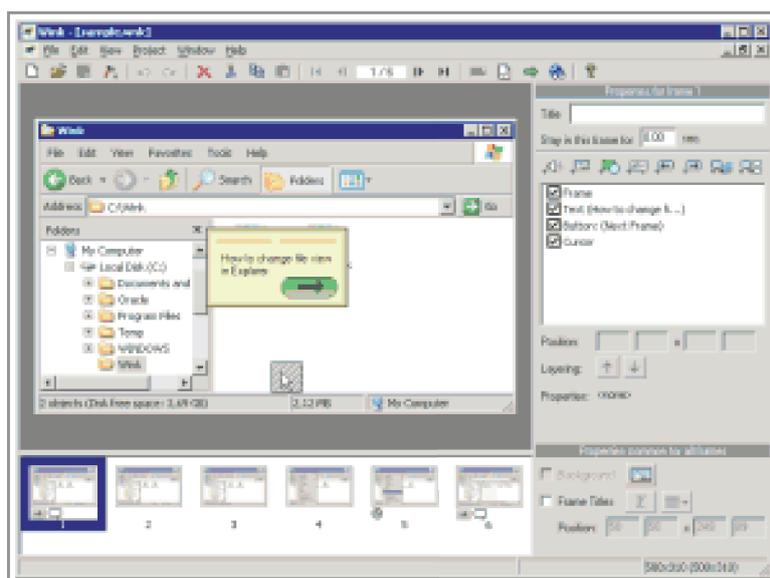


図5 Wink 利用画面

OSや動画記録形式に応じて、さまざまなスクリーンキャストの作成ツールを選択することができる。サイバー大学の受講環境はWindowsに限定されているため（注：2012年4月以降はマルチOSに対応）、学生と同じ環境でのプログラミングの様子を伝えるためにWindows上での収録ツールを利用する。いくつかのツールを試用して、プログラミング時のソースコード中の文字を読みやすいものを選択した。そこでWinkというツールを選択して利用している。Winkは、Windowsの操作画面の録画や、録画後の編集、SWF（Adobe社Flash）形式での動画出力などの機能を持つ。Winkの動画はWebブラウザ上で再生されるため、学生は特に追加ソフトウェアを必要とせずにWebブラウザ上から直接視聴することが可能である。

スクリーンキャストでは、Java言語を用いた実際のプログラミングの様子を収録して公開している。表6の12個のテーマを掲載しており、それぞれのテーマについて、実際に教員がプログラミングを開始して完成するまでの過程を収録している。ひとつのテーマ内では、作業過程を複数の動画に分けて収録している。それらのスクリーンキャストでは、全体の設計から各部の実装までの一連のプログラミングの様子を見せている。その際の思考過程についても、画面上へ文字タイピングすることによって提示している。また、実際のプログラミングの様子を知ってもらうために、プログラミング途中でのミスが発生やそ

表6 演習中で公開しているスクリーンキャストテーマ一覧

- |                      |                    |
|----------------------|--------------------|
| (1) Javaプログラムの基本について | (7) 課題プログラムの作成     |
| (2) 変数の利用例           | (8) グラフィックスプログラム   |
| (3) 制御構造と配列          | (9) http通信プログラム    |
| (4) オブジェクトの定義と利用     | (10) ファイルからの入出力    |
| (5) 数当てゲームの作成        | (11) タートルグラフィック    |
| (6) 素数計算プログラムの作成     | (12) ユーザー入力とグラフィック |

表7 スクリーンキャストの提供例

<p>「数当てゲームの作成」</p> <ol style="list-style-type: none"> <li>1. ソースファイル作成 Javaのソースファイルの作成です。</li> <li>2. プログラムの全体構成を設計～乱数生成 主クラスやmainブロックなどのJavaプログラムの基本的な構成を整えて、「数当てゲーム」に必要な基本処理を列挙しておおまかな全体像を日本語で記述してみます。さらに、ひとつめに必要になる「乱数生成」部を作成します。</li> <li>3. ユーザーからのデータ入力 ゲームのためには、ユーザーからデータ入力を受け付ける必要があります。Javaプログラム内でユーザーからのキー入力を受け付ける部分を作成します。</li> <li>4. ユーザーからのデータ入力（続き） キー入力部では「例外処理」と呼ばれる特別な処理が必要になります。例外処理を付け加えて、キー入力部を完成させます。</li> <li>5. 答えと入力値との比較・正解するまで繰り返し ユーザーからの入力値と、コンピュータが生成した乱数を比較して、数当てを行います。正解が得られるまで繰り返すという処理を加えます。</li> <li>6. 正解 / 不正解・解答表示 数当ての結果から、正解や不正解の表示を行います。</li> <li>7. 仕上げ 細かな表示部分などを修正して、完成させます。</li> </ol>
---

の訂正の様子もあえて残している。これらのスクリーンキャストを通じて、学生に対してプログラミング過程の進め方や考え方、プログラミングにおける試行錯誤を伝えることに取り組んでいる。

表7はスクリーンキャストとして提供している内容の一例である。コンピュータが乱数で生成した数を当てる、「数当てゲーム」の作成の様子を7つの段階にわけてそれぞれスクリーンキャストにて説明している。このテーマでは、プログラムの全体設計から機能拡張の繰り返しによって完成度を高めていく様子を提示している。図6は、スクリーンキャスト動画の例である。プログラミングにおける試行錯誤やエラー訂正などの様子を直接的に提示している。その際の思考過程については、画面上にタイピングすることで明示している。また、プログラミング中に生じた疑問についてWWW上から調査する過程なども見せている。プログラミング過程でのさまざまな操作はすべて理由があることなので、その理由を納得してもらうことで受講生自身が同様の過程をなぞることができるように誘導している。スクリーンキャストによって、プログラムはいきなり完成するものではなく完成に近づけていくものだということの納得や、その方法論の学習につなげている。

スクリーンキャストは受講生が自由に視聴することができるため、理解が不十分な部分を繰り返して視聴することが可能である。プログラミングの過程を直接に、そして自由に視聴することができるという点では、同期的に受講する対面型の授業では実現不可能な運営形態である。これはオンライン型演習ならではの利点である。

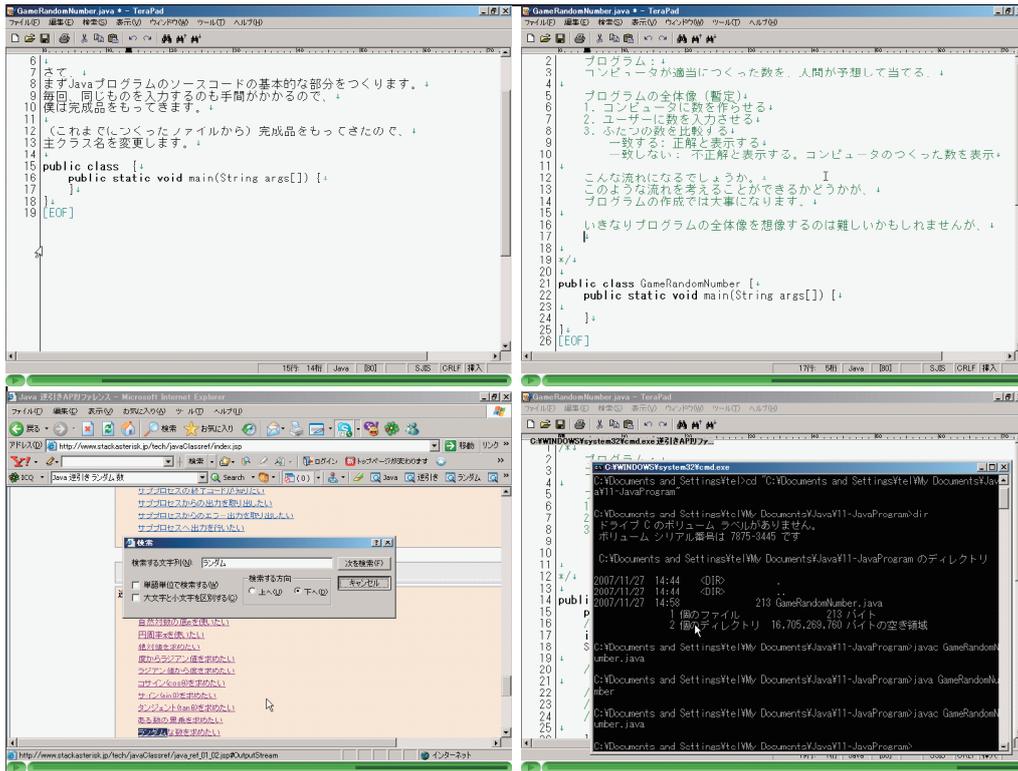


図6 スクリーンキャストの例

### 4.3. TAの活用

演習授業では、教員のアシスタントとしてTA (Teaching Assistant) が配置される。Java プログラミング演習では、TA を積極的に利用する運用体制を構築している。TA が受講生の課題について個別に対応している。これまでも特にプログラミングに不慣れな受講生については、TA が遠隔操作ソフトウェアを利用して個別対応したこともある。このような運用体制を固めていくことで、教員負担の軽減とTAへの演習知識の蓄積を目指している。これにより、TAを増やすことで今後の学生数増加にも対応可能な体制作りを意識している。

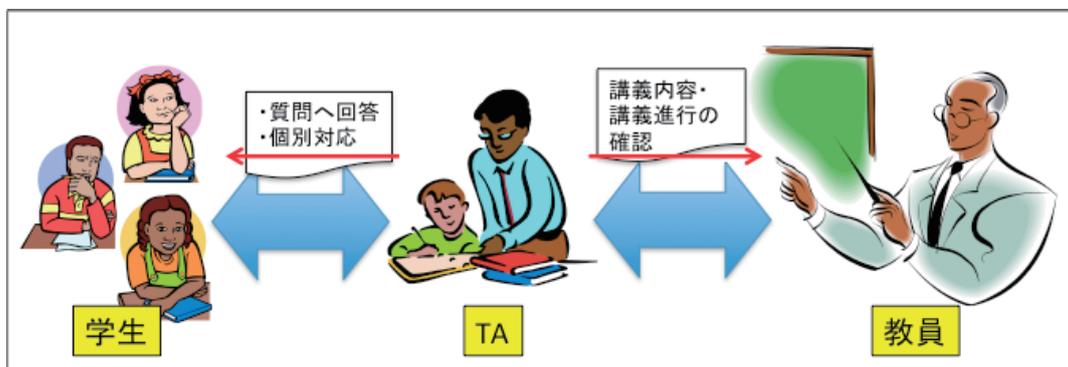


図7 TAの役割

#### 4.5. 講義進行の工夫

Java プログラミング演習では、受講生はプログラム作成に継続して取り組む。演習では、前段階で習得したことを基にして次段階の演習へと連続して取り組んでいく。そのため、演習の途中が理解出来ないとそれ以降の演習継続が困難となる。そこで、途中で脱落のないように、最低でもここまでのプログラムを作成できていれば演習継続が可能となる最低ラインを設定するようにしている。この最低ラインをスクリーンキャストでも配信することで、見よう見まねでもプログラミングを継続できるように工夫している。

ただし、演習全体の目標となるプログラミング能力の習得のために受講生はそれぞれ異なる内容のプログラムを最終成果物として提出するように定めている。このときは、受講生のレベル差にも配慮して難易度の異なる複数のテーマ例を提示することになっている。初級者にはこれまで学習してきたプログラムの改造に取り組んでもらい、上級者には自由なテーマ設定を許している。これにより、初級者には入門レベルのプログラミング能力の習得を達成してもらいつつ、上級者には自身がこれまでに得ている能力も合わせつつ高度な課題へ挑戦する自由度を実現している。TA を含む指導体制は、この自由度の高さの実現に費やしている。

### 5. ま と め

本稿では、著者が担当する Java プログラミング演習を例として、サイバー大学における演習講義の取り組みについて紹介した。通常の講義と異なり、受講生側での課題の実施が必要となる演習では、受講生が円滑に課題を実施するための支援が必要となる。特に、プログラミングではプログラム言語の知識という形式知に限らず、プログラムを作成する際の考え方という暗黙知も必要となる。Java プログラミング演習においても、プログラミング能力の学習のためにその両面的な知識の獲得が必要になる。

Java プログラミング演習の運営にあたっては、いくつかの工夫を施している。本稿では、教科書の設定、スクリーンキャストの提供、TA の活用について報告した。プログラム言語の仕様や命令などの誰が教えても同じ内容となる形式知については教科書を用いて、受講生共通での知識獲得を図っている。そして、プログラミングにおいて最も重要となるプログラム作成の考え方というノウハウ的な暗黙知については、実際にプログラムを作成する過程をスクリーンキャストとして提供することでの伝達を図っている。実際の過程を自由なペースで視聴しながら暗黙知について学習することができるのは、オンライン型演習ならではの可能性がある。これらの取り組みのうえで、TA を補助として積極的に前面にだすことで、人手が必要な個別対応の余力につなげている。これらの取り組みによって、形式知から暗黙知、そして例外处理的な個別対応までを実現している。

参考資料

[Cyber-U] サイバー大学, <http://www.cyber-u.ac.jp/>, 2011/12/15

[BookJava] 改訂 新 Java 言語入門 (ビギナー編) 林晴比古 SoftBank Creative ISBN : 4-7973-3021-X

[SCast] Wikipedia : スクリーンキャスト, <http://ja.wikipedia.org/wiki/%E3%82%B9%E3%82%AF%E3%83%AA%E3%83%BC%E3%83%B3%E3%82%AD%E3%83%A3%E3%82%B9%E3%83%88>, 2011/12/15

[Wink] Wink 配布サイト, <http://www.debugmode.com/wink/>, 2011/12/15

[WMEnc] Microsoft Media Encoder, [http://www.microsoft.com/expression/products/EncoderPro\\_Overview.aspx](http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx), 2011/12/15