

# C#プログラミング教育における反転授業の実施

小島篤博

大阪府立大学 大学院 人間社会システム科学研究科

## A Practice of Flipped Classroom for Teaching C# Programming Language

Atsuhiko Kojima

Graduate School of Humanities and Sustainable System Sciences,  
Osaka Prefecture University

Introducing a style of flipped classroom into a course of learning programming language bring the benefits that students can make more exercise in the class time by learning basic syntax and algorithms in advance. In this paper, a practice of flipped style education on a programming course of C# for the first-year students is reported that consists of short movies for preparation and assurance tests. An analysis and evaluation on the practice are also presented.

キーワード: プログラミング教育, 反転授業, C#, 予習ビデオ, 学習管理システム

### 1 はじめに

大学の情報系課程におけるプログラミング教育では、問題解決のための基本的な考え方や、それを具体化するためのプログラミング言語の文法の2つを習得することを大きな目標としている。一般にプログラミングを修得するために、自分でプログラムを作成・コンパイルし、動作を確認するという一連の過程を繰り返す演習が不可欠である。このため多くの授業では、プログラミングの考え方や文法を教師が解説し、写経型の課題を行った上で、より応用的な課題に取り組むといった方式が取られている。しかしながら、授業の時間的な制約のため、応用的な課題については個々の学生が時間外に行わざるを得ず、教師のサポートを受けにくいという問題を抱えている。

一方、近年アクティブ・ラーニングの一手法として、授業と時間外の自習との役割を逆転した反転授業の導入が試みられている<sup>(1)(2)</sup>。反転授業の長所としては、事前に学習内容をビデオ等の自習教材で予習することで、授業では演習を中心とした能動的な活動に多くの時間を割り当てることができるという点が挙げられる。この利点は、前述のように演習の比重が大きいプログラミング教育においても効果的であると考えられ、大学のプログラ

ミング科目における実践例も報告されている<sup>(3)(4)</sup>。

本論文では、大阪府立大学の情報系課程（現代システム科学域 知識情報システム学類）におけるプログラミング科目として、1年次後期に担当されている「プログラミング入門」において、反転授業用の予習ビデオ教材や予習確認のための小テスト等の教材を開発し、授業実践を行なった事例について報告する。

### 2 プログラミング科目の概要

まず、授業実践の対象とした科目「プログラミング入門」の授業目標および授業計画について説明する。

#### 2.1 授業目標

授業目標としては、問題解決のための手法として基礎的なアルゴリズムを理解し、C#の基本的な文法とプログラムの作成方法を習得することである。ただし、科目を担当している課程は情報系ではあるが工学系ではなく、将来的にはプログラム開発技術者と言うより、広くシステム構築の企画・運営に携わる職種を目指す学生が多い。

同科目においては、前年度（2017年度）まではC言語を使用言語として設定していた<sup>(4)</sup>。しかしながら、研

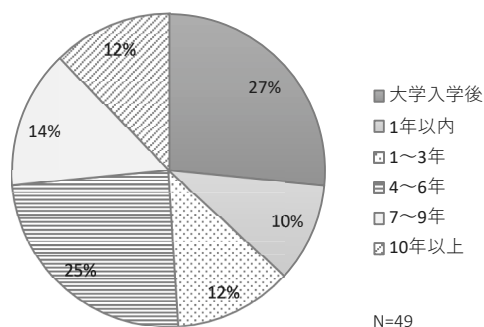


図 1: パソコン使用経験

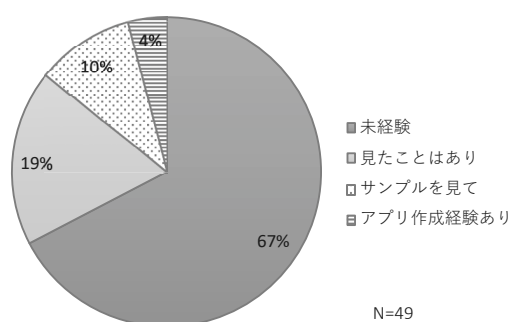


図 2: プログラミング経験

研究室配属後に実際に C 言語を使うケースは少なく、また C 言語では配列やポインタ、メモリ管理の方式が時代遅れとなってしまうこと、クラスライブラリの利用などでオブジェクト指向の基礎的な知識が不可欠になっていることなどから、新しい言語への移行を検討した結果、以下の理由から C#が最適であると判断した。

- 各研究室で主に使用するプログラミング言語を調査したところ、最も多く使用されている言語が C#と Python であった (同数)。なお、他には Java、Swift、R などがあった。
- 多くの学生にとっては、この科目で学ぶのが初めてのプログラミング言語であり、できるだけ他の言語と共通性が高いことが望ましい。C#は C++、Java、Swift と類似点が多く、一旦 C#を学んでおけば他の言語への切り替えも難しくないと考えられる。一方、Python は独特な部分が多く、他の言語への移行は苦勞が予想される。

よって 2018 年度からは C#を採用することとなり、従来の C 言語用の教材は一部を除いて再利用ができず、新たに作成し直した。

## 2.2 授業計画

授業計画の参考とするため、授業開始時に受講者のプロフィールを把握するためのアンケートを実施した。図 1 は、これまでのパソコン使用経験を尋ねたものであり、最も多いのが「大学入学後に初めて利用した」であり、27%であった。「1 年以内」と合わせると 37%がパソコン使用経験が 1 年以内となっている。本学では 1 年次前期

に共通教育として情報リテラシー教育を実施しており、パソコン未経験者はいない<sup>(5)</sup>。

また、図 2 はプログラミング経験について尋ねたものであり、「全く経験がない」(67%)、「プログラムを見たことはあるが、自分で試してみたことはない」(19%)を合わせると、86%がプログラミングの経験がないと回答している。プログラミング経験については年度によって変動はあるものの、大半の学生が未経験であり、授業としては初めてプログラミングを行う初学者であることを想定する必要がある。

これらに基づいて、初めて C#を学ぶ学生が一通りの文法を習得し、プログラムが作成できることを目標とした授業計画を表 1 に示す。教科書は桑井康孝『猫でもわかる C#プログラミング 第 3 版』<sup>(6)</sup>を採用した。

各回の授業は基本的に次のような手順で進める。

1. 予習ビデオの視聴  
授業日の 1 週間程度前から、その回の内容を予習するためのビデオを公開する。受講者には授業開始前までに視聴することを指示する。
2. 予習確認テスト  
授業の最初に、予習ビデオの内容を確認する Web 上の小テストを実施する。出題は 2～4 問程度で、問題プールからランダムに出題する。得点は成績にも反映することを通知しておく。
3. プログラム作成演習  
その回の学習内容に対応したプログラムの作成を演習形式で行う。受講者は教師に質問したり、受講者同士で相談したりしながら進める。
4. 課題提出

表 1: 「プログラミング入門」の授業計画

|    | 項目               | 内容  |
|----|------------------|---|
| 1  | イントロダクション        | C#の概要,<br>プログラムの作成手順                      |
| 2  | C#プログラムの構造       | クラス・メソッド,<br>基本的な文法, 変数                   |
| 3  | 変数とデータ型          | 変数の宣言, 入出力,<br>基本的なデータ型                   |
| 4  | 文字列と演算子          | 文字列の操作,<br>算術演算子                          |
| 5  | 演算子と計算           | 型変換, 関係・論理演算子,<br>演算子の優先順位                |
| 6  | 条件分岐 (1)         | if 文, 多分岐                                 |
| 7  | 条件分岐 (2)         | switch 文, インデント                           |
| 8  | 中間テスト            | 1~7 回の内容,<br>エラー・デバッグ                     |
| 9  | 反復処理 (1)         | for 文, 制御変数,<br>1~n の和, 乱数, 数当て           |
| 10 | 反復処理 (2)         | while 文, 多重ループ,<br>数値計算, break/continue 文 |
| 11 | 配列               | 配列の宣言・初期化,<br>多次元配列                       |
| 12 | 反復処理と配列          | ジャグ配列, foreach 文,<br>素数の判定, 組合せ探索         |
| 13 | クラス              | クラスとインスタンス,<br>インスタンスの生成                  |
| 14 | 期末テスト<br>メソッドの定義 | 8~13 回の内容,<br>データの授受, データ隠蔽               |
| 15 | 総合演習 (補講)        | バブルソート                                    |

指定された課題を 1 週間後までに提出させる。課題の説明と指示は授業中に行い、取り組む時間を与える。一旦提出した課題で間違いなどがあれば指摘し、更に 1 週間の期限で再提出させる。

なお、第 1 回は初回であるため、授業の進め方についてガイダンスを行い、予習確認テストは実施しない。中間・期末テストは授業の一部として実施し、テスト終了後は通常の授業を続けるものとした。ただし、休講の関係から第 15 回が補講となり、平常とは異なる日時で実施した。このため第 15 回については予習ビデオを作成していない。

### 3 反転授業教材の開発

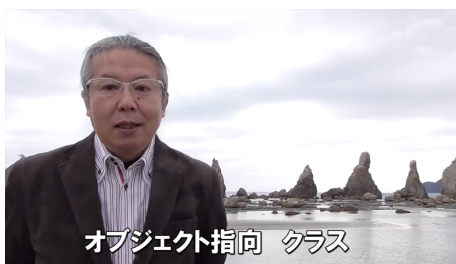
前章で説明したプログラミング科目を反転授業で実施するための教材として、予習ビデオと確認テストを開発した。大阪府立大学では、すべての開講科目で利用可能な公式サービスとしてオープンソースの LMS である Moodle を運用しており、これらの教材も Moodle 上で提供している<sup>(7)(8)(9)</sup>。

表 2: 予習ビデオ一覧

|      | 形式      | 内容                  |
|------|---------|---------------------|
| 01-1 | 04:28 B | ガイダンス, 導入           |
| 01-2 | 06:28 B | C#の概要               |
| 02-1 | 01:33 A | 授業の概略               |
| 02-2 | 09:29 B | プログラムの構成            |
| 02-3 | 06:45 B | 変数                  |
| 02-4 | 04:15 B | 変数の入出力              |
| 03-1 | 01:54 A | 授業の概略               |
| 03-2 | 03:49 B | データ型                |
| 03-3 | 06:15 B | 整数型                 |
| 03-4 | 07:14 B | 実数型・文字列型            |
| 04-1 | 01:51 A | 授業の概略               |
| 04-2 | 04:02 B | 文字列とは               |
| 04-3 | 05:13 B | 文字列の操作              |
| 04-4 | 05:48 B | 算術演算子               |
| 05-1 | 01:38 A | 授業の概略               |
| 05-2 | 05:20 B | 型変換, フォーマット出力       |
| 05-3 | 05:40 B | 様々な演算子              |
| 05-4 | 02:58 B | 演算子の優先順位            |
| 06-1 | 01:25 A | 授業の概略               |
| 06-2 | 06:32 B | if 文の基本形            |
| 06-3 | 02:00 B | if 文による多分岐          |
| 06-4 | 03:53 B | コーディングスタイル          |
| 07-1 | 01:12 A | 授業の概略               |
| 07-2 | 02:45 B | インデントの意味            |
| 07-3 | 06:18 B | switch 文            |
| 07-4 | 03:38 B | if 文と switch 文の違い   |
| 08-1 | 01:26 A | 授業の概略               |
| 08-2 | 05:28 B | コンパイルエラー            |
| 08-3 | 03:52 B | 実行時エラー, デバッグ        |
| 09-1 | 02:09 A | 授業の概略               |
| 09-2 | 06:00 B | for 文の基本形           |
| 09-3 | 06:04 B | 様々な繰返し条件            |
| 10-1 | 01:29 A | 授業の概略               |
| 10-2 | 06:02 B | 多重ループ               |
| 10-3 | 06:57 B | while 文, do-while 文 |
| 10-4 | 02:18 B | ループの中断とスキップ         |
| 11-1 | 01:58 A | 授業の概略               |
| 11-2 | 06:18 B | 配列                  |
| 11-3 | 06:19 B | 多次元配列               |
| 12-1 | 01:10 A | 授業の概略               |
| 12-2 | 03:58 B | ジャグ配列               |
| 12-3 | 02:18 B | foreach 文           |
| 12-4 | 05:25 B | 演習課題 素数の判定          |
| 12-5 | 02:52 B | 演習課題 組合せの探索         |
| 13-1 | 02:31 A | 授業の概略               |
| 13-2 | 04:50 B | オブジェクト指向            |
| 13-3 | 05:24 B | クラスの定義              |
| 13-4 | 03:14 B | new 演算子, 値型と参照型     |
| 14-1 | 01:52 A | 授業の概略               |
| 14-2 | 06:35 B | メソッドの定義             |
| 14-3 | 02:17 B | クラス定義の例             |

#### 3.1 予習ビデオ

今回予習ビデオとして作成したものの一覧を表 2 に示す。形式は A, B の 2 種類からなり、A は教師がその回の概略を実写形式で口頭で説明したもの、B は PowerPoint



**インスタンスの生成**

- 定義されたクラスの実体であるインスタンスを生成するためには **new 演算子** を用いる。

```
Complex a; // Complex型の変数aを宣言
a = new Complex(); // インスタンスを生成、aに代入
```

- newは、指定したクラスの新たなインスタンスを生成する働きをする。newしていない時点では、まだインスタンスは存在せず、変数は**空の状態**。

```
Complex a;           a = new Complex();
```

図 3: 予習ビデオの例

のスライドに音声のみを吹き込んだものである。それぞれ再生時間があまり長くなり過ぎないように、7分を上限の目安として作成した。図3に予習ビデオの例を示す。

反転授業では、受講者は毎回授業前にビデオを視聴し理解しておくことが求められるが、一般にスライドに音声吹き込んだもののみでは単調となってしまうがちであり、飽きさせずに視聴を促す工夫が必要である。本研究では、香川大学の林が実践している事例に倣い<sup>(10)</sup>、予習ビデオの本編であるスライド形式のビデオに先立ち、教師がその回の概略を説明するという態で毎回異なる場所を訪問し収録を行なった。

スライド形式のビデオは、通常内容を説明する PowerPoint 形式の資料に音声を収録し、ビデオ出力したものを動画編集して作成した。音声の収録は説明用の原稿をあらかじめ用意し、スライドごとに音声を吹き込むという手順で行なった。当初は原稿を用意しない形で収録を試みたが、何度もリテイクすることになり結果として収録時間が長引いてしまうという経験に基づいている。

### 3.2 確認テスト

毎回の授業で予習ビデオの内容を確認するための小テストを作成した。授業のガイダンスを行なった初回、中間・期末テスト、補講の4回分を除く11回分であり、内容は予習ビデオで説明した中から出題している。問題は Moodle で作成可能な多肢選択、短文記述、穴埋め、組

配列の記述について、空欄を埋める最も適切な単語を、同じ色の選択肢からドラッグ&ドロップして埋めよ。

次のような配列を宣言したとする。

```
int[] data = new int[40];
```

この配列の要素数は  から  である。

一般に、要素数Nの配列では、添字の範囲は  から  となる。

39  41  40  0  1

N-1  1  0  N  不定

チェック

配列の各要素が以下の値になるよう、初期化せよ。

int型の配列a

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 4 | 8 |
| 0 | 1 | 2 | 3 |

解答:

チェック

図 4: 確認テストの例

合せ、正誤の中から問題ごとに選択し、それぞれ類似問題を3~5問程度用意してランダムに出題できるようにしている。図4に出題例(第11回)を示す。

また、予習確認テストとは別に、第8回と14回にそれぞれ中間・期末テストを作成した。これは、前半と後半の内容の知識定着度を判定するために実施するものであり、それぞれの範囲から出題している。問題自体は予習確認テストと同様の形式であるが、重複はしていない。出題数は、予習確認テスト34問、中間・期末テスト15問、ランダム出題のために用意した問題バンクの総数は211問である。

## 4 授業実践と評価

以上の教材を利用し、授業での実践を行なった。対象科目は、大阪府立大学の現代システム科学域 知識情報システム学類1年次後期配当の「プログラミング入門」であり、毎週金曜5コマ(16:15~17:45)、期間は2018年9月28日~2019年2月2日の計15回である。科目の受講者数は56名、このうち学習履歴の研究利用に同意した49名を対象に、以下の分析を行なっている。

### 4.1 ビデオの視聴行動

まず、予習ビデオの視聴行動について、Moodleのログから統計を取った。図5は、各回のビデオ視聴率(受講者数に対するユニーク視聴者数の割合)の推移を示す。当初、特に2~5回目までは8割の受講者が視聴した上で授業を受講していたが、6回目以降はほぼ6割程

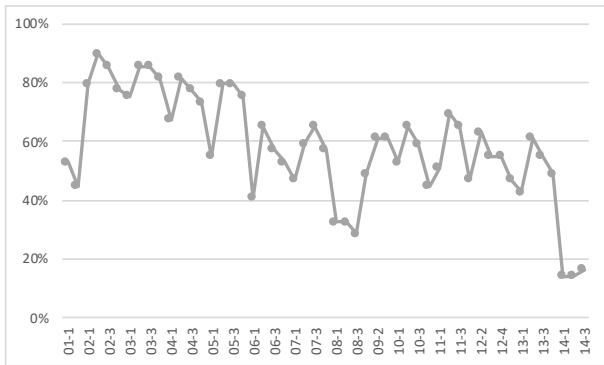


図 5: ビデオ視聴率の推移

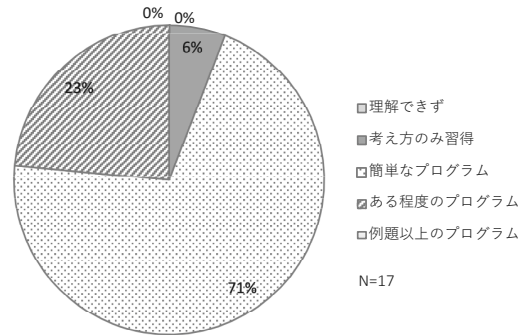


図 7: C#の習得レベル

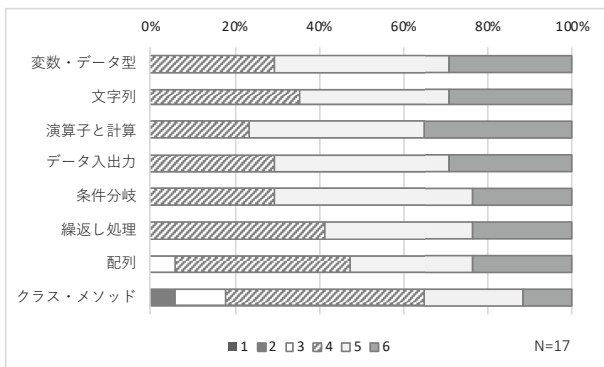


図 6: 項目ごとの達成度

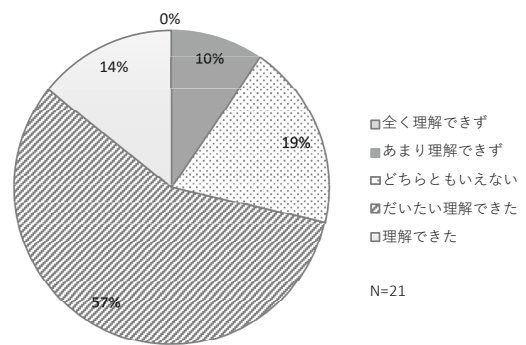


図 8: 各回の概要説明で目標が理解できたか

度で推移している。14 回目が落ち込んでいるのは、期末テストのため通例の予習確認テストを行わず、試験範囲の復習に時間が充てられたこと、他の科目の試験期間に重なったことが要因と考えられる。

また、ビデオの形式としては、各回とも実質的に内容を説明する B 形式のビデオ (PowerPoint + 音声、60.5%) に比べ、実写映像で概要を説明する A 形式の方が視聴率は低かった (52.9%)。

## 4.2 アンケート調査

毎回の授業後に学習項目ごとの理解度・達成度について、また初回と第 15 回には、科目受講前および受講後の全般的な目標や達成度、予習ビデオに関するアンケートを実施した。図 6 は、C#の大きな学習項目ごとに達成度を自己評価してもらった結果である。評価は「1. 全く達成できていない」から「6. 完全に達成できた」までの 6 件法で行なった。結果として、すべての項目で 80% 以上の受講者が達成できた (4~6) と回答しており、特に配列とクラス以外の項目では全員が 4 以上となっている。ただし、授業日程の関係からアンケートを実施したのは補講日であり、回答者は 17 名である。

また、授業を終えた時点での C#の習得レベルを、「ほとんど理解できず、身につかなかった」から「自分でも C#について積極的に勉強し、授業の例題以上のプログラムが作成できるようになった」の 5 段階から選んでもらったものを図 7 に示す。最も多かったのは、「C#の基本的な文法は習得し、簡単なプログラムが作成できるようになった」(70.6%)、次いで「授業の内容はほぼ理解でき、ある程度のプログラムが作成できるようになった」(23.5%) であった。

以上の結果から、基礎的なプログラミング科目としてはほぼ目標を達成していると考えている。

次に、今回反転授業の教材として導入した予習ビデオに関して、実写による概要説明で、その回の目標が理解できたか (図 8)、スライド資料の部分にも、音声だけでなく教師の実写映像があった方がよいか (図 9) を尋ねた。

各回授業の概略説明のために作成した教師による実写映像については、目標が理解できたとの回答が合わせて 71.4% であり、効果はあったと考えられるが、4.1 でも述べたように内容のビデオに比べて視聴率は低く、アンケートからはその原因はわからなかった。

今回作成した予習ビデオでは、本編は教師の映像はな

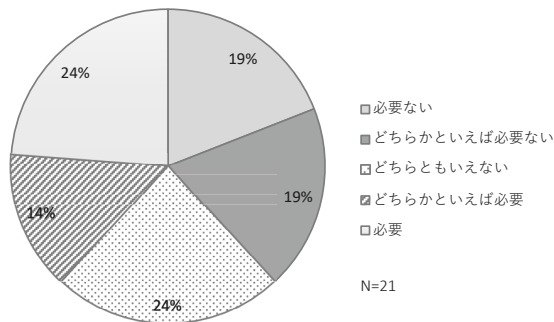


図 9: 教師の実写映像はあった方がよいか

く音声のみの説明とした。実写映像は、編集時に映像の繋ぎ目で不連続になるなど、品質の高い教材を作成するには少なくない手間と時間を要する。大学の通常の講義で継続的にビデオ教材を準備する場合、作成の手軽さと映像の品質はトレードオフの関係にあり、スライド資料に音声を取録する形式のものはコストパフォーマンスに優れていると考えられる。教材を視聴した受講者の意見では、教師の実写映像の有無は図 9 に示す通り大きく回答が分かれる結果となった。現在、実写映像の代わりに CG のキャラクタを使用して、撮影・編集の手間を抑えて映像を作成する方法を検討している。

その他、記述回答として「最初の実写の概要説明があって楽しく見れてよかったです」のように評価する感想がある一方、「頭に入れたことをなぞるだけなので授業時間中の実習がそれほど楽しくなかった」のような改善提案も見られた。

### 4.3 成績との相関

最後に、予習ビデオの視聴行動と成績との相関を分析した。成績としては、プログラム作成課題、予習確認テスト、中間・期末テストをそれぞれ 100 点ずつとし、合計したものを用いた。これは最終的な成績とは異なる。また、各回のビデオを 1 度でも視聴したかどうかを累積したものを視聴数とした。すべてのビデオを視聴した場合は 51 である。図 10 に、成績と視聴数の相関を示す。相関係数は 0.65 であり、視聴行動と成績との間に正の相関、すなわちより多くの予習ビデオを視聴した学生ほど成績が良かったことが示唆された。

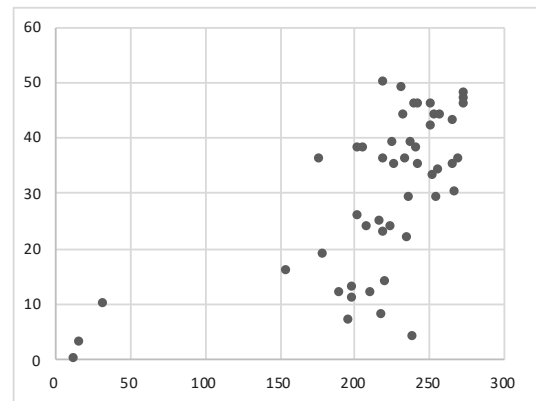


図 10: ビデオ視聴数と成績との相関

## 5 まとめ

大学の情報系課程におけるプログラミング科目において、反転授業のための教材として予習ビデオを開発し、授業による実践と評価を行なった。その結果、基礎的なプログラミング能力を修得するという目標についてはある程度達成することができた。一方、反転授業の効果としては、従来型の授業との比較などを含め、今後も検討して行く必要がある。

## 参考文献

- (1) ジョナサン・バーグマン, アーロン・サムズ, (山内祐平 訳): “反転授業”, オデッセイコミュニケーションズ (2014)
- (2) 重田勝介: “反転授業 ICT による教育改革の進展”, 情報管理 Vol.56, No.10, pp.677-684 (2013)
- (3) 喜多一, 岡本雅子: “写経型プログラミング学習と反転授業”, システム制御情報学会研究発表講演会講演論文集 60, 4p (2016)
- (4) 小島篤博: “C 言語プログラミング教育における反転授業の実践”, 教育システム情報学会 2016 年度 特集論文研究会 (2017)
- (5) 小島篤博, 真嶋由貴恵, 宮本貴朗, 青木茂樹: “e ラーニングを導入した全学情報教育における教育実践と質保証”, 教育システム情報学会研究報告 27(7), pp.213-218 (2013)
- (6) 糸井康孝: “猫でもわかる C# プログラミング 第 3 版”, SBクリエイティブ, 東京 (2017)
- (7) 小島篤博, 青木茂樹, 宮本貴朗: “大学基盤システムと連携した Moodle による授業支援システムの構築”, 日本教育工学会 第 28 回全国大会, pp.229-230 (2012)
- (8) 青木茂樹, 小島篤博, 星野聡孝, 宮本貴朗: “出席管理システムの開発・運用と利用状況解析”, 電子情報通信学会論文誌 (D), J97-D(5), pp.1053-1057 (2014)
- (9) 小島篤博, 青木茂樹, 宮本貴朗: “大阪府立大学における Moodle のバージョンアップ”, 日本 Moodle 協会全国大会 (2015) 発表論文集, pp.44-47 (2015)
- (10) 林敏浩: “e-Knowledge コンソーシアム四国の教育クラウド運用と Learning Analytics への課題”, コンピュータ & エデュケーション, Vol.38, pp.49-54 (2015)