

プログラミング教育における反転授業の実践と評価

Practice and Evaluation of Flipped Classroom in Computer Programming Education

林 康弘, 深町 賢一, 小松川 浩

Yasuhiro HAYASHI, Ken-ichi FUKAMACHI, Hiroshi KOMATSUGAWA

千歳科学技術大学 理工学部

Faculty of Science and Technology, Chitose Institute of Science and Technology

Email: {yasuhiro,k-fukama,hiroshi}@photon.chitose.ac.jp

あらまし：プログラミングでは、学生は、制御フロー表現、文法の正確さ、アルゴリズム、という知識と技能を一体として習得する必要がある。この効果的な教授に向けて、我々は2013年度より授業での知識定着・確認を図る確認テストや知識活用・創造を図るアクティブラーニングのための反転授業を実践してきた。今回、この実践を通じて得られた課題を踏まえて反転授業の実施手順を改善し、その評価を行った。

キーワード：コンピュータプログラミング、反転授業、アクティブラーニング、eラーニング

1. はじめに

本学では2013年度より学部2年生を対象にコンピュータプログラミングの必修科目「プログラミングスキル（春学期）」と「オブジェクト指向プログラミング（秋学期）」においてそれまでに開発したeラーニング教材を用いて反転授業を実践している。プログラミングスキルではC言語、オブジェクト指向プログラミングではJava言語を取り扱う。毎週の授業前に学生にeラーニング教材を予習として割り当てる反転授業により、教員は学生全体に対して自学自習に適した学習環境と技能習得に向けたアクティブラーニングを提供し、さらに個々の学生に対して学習指導を行っている。成果として本取り組みにより学生の間層の学力向上に寄与できることを確認した[1, 2]。

しかし、反転授業を実践する中で以下の課題も生じた。2014年度における取り組みの目的は、これらの課題を踏まえて反転授業の実施手順を改善し、今後、反転授業とアクティブラーニングによるプログラミング技能習得に向けた教授方法の確立と学習支援の仕組みを構築するための要件の抽出にある。

(1) 学生が予習した内容をどれくらい理解できたかその時に確認できなかった。今回、学生のニーズに応えるために予習用確認問題を設定した。

(2) 教材内容が分かりにくい、授業進度を変更する等の場合、教員が次の授業前までに予習用教材の内容を容易に変更できなかった。今回、教材作成に用いたPowerPointデータから動画を生成し対応した。

(3) 実習時に他の学生との協調作業を伴うアクティブラーニングの内容が充実していなかった。今回、この実施方法を改善した。

2. 本取り組みにおける反転授業とアクティブラーニングの実践方法

本取り組みでの実施手順は次の通りである。

1. 教員は授業進度に合わせて、PowerPoint からMPEG4形式の動画教材を生成し、eラーニング

上で閲覧できるようにする

2. 教員はeラーニング教材を用いて学生にその週に学習する内容を予習として割り当てる
3. 学生は予習を行い、確認問題を解く
4. 授業では、開始直後に教員は出席確認と予習の理解度を図るための確認テストを実施する
5. 授業前半、教員は約15分程度で学習内容を説明する。説明が長い場合、説明を2回に分ける
6. 学生はその週の内容の課題に取り組み、教員は個別指導と課題確認カードの配布を行う
7. 課題確認カードは、学生が課題終了後にTAによる課題の確認を経て、回収される。出席確認と課題確認カードが揃って出席と見なされる。課題が授業中に終わらない場合、翌週までの宿題となる
8. 授業後半から次週までに、学生はグループに分かれて協調学習に取り組む。1グループは3~4名で構成される。毎週、グループのリーダーを変更し、リーダーはその週の活動報告を行う
9. 授業時間外の自主学习で質問がある場合、時間外TAが大学にて対応する

穴埋め形式の予習用確認問題にはヒントと関連教材へのリンクがあり、学生が回答できない場合、再度予習を促すようにしている。授業で学生に提示される一連の問題の難易度は教員によって学生の学習意欲を考慮し、予習用確認問題は易しく、確認テストは難しく、実習課題は程良く、設定されている。

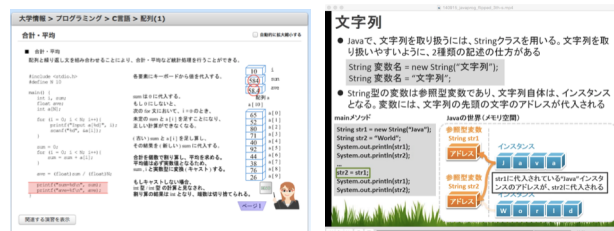


図1 eラーニング教材 (左: Flash, 右: 動画)

使用教材はPowerPointにて原稿を作成され、Flash

を用いて e ラーニング教材化される (図 1 左). この変換作業は学内の情報教育部門が担当する. 今回, 「オブジェクト指向プログラミング」では, 教員が PowerPoint のナレーション機能により解説を録音し, PowerPoint 2013 から対応された MPEG4 形式でのファイル出力により動画を作成し, e ラーニングにアップロードした (図 1 右). これにより授業進度に合わせた教材内容を自由に変更可能とした.

実習では, 教員が提示する仕様書に従って実装を行うだけでなく, グループでサンプルプログラムの内容理解を図るための議論, 自分たちのソフトウェアの設計と実装, 作成したソフトウェアのプレゼンテーションと仕様書作成をアクティブラーニングとして行わせた.

3. 実践結果と考察

プログラミングスキルとオブジェクト指向プログラミングはいずれも 2 コマ/週で, 履修者数はそれぞれ 49 名と 59 名 (再履修者含む) であった.

3.1 予習用演習問題の学習状況

プログラミングスキルでは, 10 週にわたって反転授業を実施した. その際, 新たに設定した予習の成果を学生自ら確認する問題 (予習用演習問題) に学生が取り組んでいるか確認した (図 2). 達成度 P_i は一人の学生 i が出題された問題数 (Q) に対していくつ正解したか (A) を表し ($P_i = A/Q \times 100$), 平均達成度 $\bar{P}$ は履修者全体 N での達成度の平均を採ったものである ($\bar{P} = 1/N \times \sum P_i, i=1..N$). 平均達成度が高ければ, 履修者全体が予習を行い, 予習の時点で予習用演習問題を通じて最低限の知識を身につけたと見なす. 約 8 割の平均達成度で推移していることがわかる. 学生は予習を継続し演習問題に取り組んでいることが分かる. また併せて, その週の授業冒頭に実施した確認テストの平均点も示す. 確認テストは授業を通じて最終的に理解すべき全ての内容が含まれており, 予習用演習問題よりも難しく設定されている. 第 7 週以降, 確認テストの平均点が低くなっているのは学習内容が関数, 構造体, ポインタと抽象的理解が必要となるためと考えられる.

3.2 動画教材の生成と学生の反応

オブジェクト指向プログラミングでは, 15 回の授業のうち, 教員が授業進度を調節した回数は 5 回あり, 予習教材の内容を変更した. 遅延なく予定された日時で予習教材を提示できた回数は 3 回であった. 残り 2 回は, 新規にスライド作成しなければ対応できない内容, 動画変換にトラブルがあったため, 予定の日時で予習教材を提示できなかった.

履修学生 59 人に動画教材に関してアンケートを実施した. 設問「教材は内容の理解に役立ちましたか?」に対して, 強くそう思う 5 人, ややそう思う 16 人, どちらとも言えない 8 人, あまりそう思わ

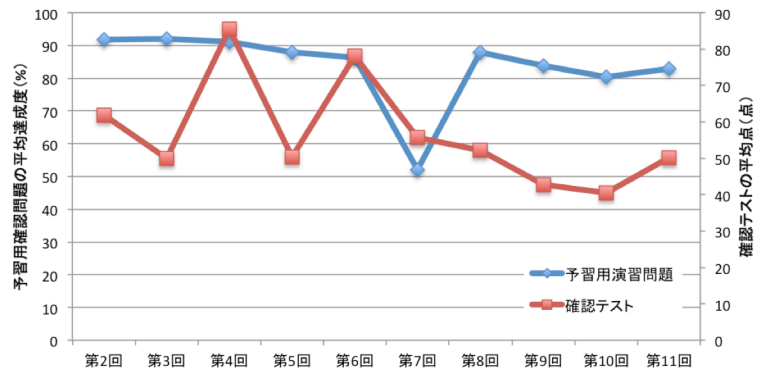


図 2 予習用演習課題の平均達成度と確認テストの平均点

い 7 人, 全くそう思わない 0 人 (有効回答数: 36) という結果となった.

3.3 アクティブラーニングへの学生の反応

プログラミングスキルの履修者 49 名に対して, 反転授業とアクティブラーニングに関するアンケートを実施した (有効回答数: 22). 「反転授業を採用した場合と採用しない場合を比較して, どちらがプログラミング技能を高めるのに役立つか?」という問いに対して, 22 名全員が「反転授業を採用した授業」と回答した. またそれ以外に得られた有益なコメントは次の通りである.

- 個人だと面倒臭がってやらなかったり, プログラミングがわからず手が進まない人でもグループ内で聞きあったり, 先生や TA さんに相談して教えてもらうことで理解も深まりよいと思った
- 高度なものを作る, 欠点を無くすならばもう少しアクティブラーニングの時間を増やすべき
- 実際に行うことにより対応力や経験力が高まる
- 次週のやることが明確にわかるので今週どの程度復習と予習に時間を費やせばよいかわかる

4. まとめと今後の課題

2013 年度より実践しているプログラミング教育における反転授業の取り組みの中で生じた課題を踏まえて, 2014 年度はその実施手順を改善し評価を行った. 今後は, 得られた知見に基づき, 反転授業とアクティブラーニングによるプログラミング技能習得に向けた教授方法の確立と学習支援の仕組みの構築を図る.

参考文献

- (1) 林 康弘, 深町 賢一, 小松川 浩: “e ラーニング利用による反転授業を取り入れたプログラミング教育の実践,” 社団法人私立大学情報教育協会, ICT 利用による教育改善研究報告第 16 巻, pp.19-23, 2013 年 11 月.
- (2) Y. Hayashi, K. Fukamachi and H. Komatsugawa: “Collaborative Learning in Computer Programming Courses that Adopted the Flipped-classroom,” The 3rd International Conference on Learning and Teaching in Computing and Engineering (LaTICE2015), April 2015.