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

Practice Approach of Computer Programming Education that Adopted the Flipped Classroom by Using e-Learning

林　康弘　深町賢一　小松川 浩
千歳科学技術大学総合光科学部

Abstract: We present the practical approach of computer programming education that adopted the flipped-classroom by using e-Learning. The main feature of this approach is to use most of the time usually dedicated to lecture for practicing by assigning the students the e-Learning materials as preparation before the actual class. We gave the students homework that involved studying the vocabulary and the grammar of the C programming language. This decreased the time a teacher spent lecturing. In addition, the students were assigned basic and applied questions that related to the practical programming in the class. Our goal is to improve the degree of understanding content by clarifying difficult concepts or questions generated before the actual class. We hope to create a learning cycle that encourages students to prepare for the class appropriately and use the class time to make practical software. We confirmed the effectiveness of this approach by comparing examination scores from last year to this year. Performance showed in improvement in those students who achieved a mid to high level grade on examinations. Additionally, 54% of the students answered that the flipped-classroom was good and the computer programming was fun in questionnaires. The detail of this approach is shown in this paper.

Keywords: computer programming, flipped classroom, e-Learning, ICT, blended learning

1. はじめに

情報通信技術の発達により社会が変化する中，学生の論理的思考力と問題解決能力を養う方法として，プログラミングは有用である．

今年度春学期，科目「プログラミングスキル」の履修者88名を対象に反転授業（Flipped Classroom）を行った．情報系教員３名，TA（Teaching Assistant）８名が担当した．本取り組みの特徴は，ブレンド型授業にて用いるｅラーニング教材を予習に用いることにより，講義時間の多くを実習に使うことを可能にした点である．プログラミング言語の文法解説を予習に置き換えることにより，講義での教員の解説時間が短くなる分，実習での文法の練習課題だけでなく，確認テストと実用的なプログラムを書く課題に取り組ませた．本取り組みの目的は，①学生が予習段階で分からない箇所を明確にし，授業内容の理解度の向上を図ること，②実習時間を有効活用し，学生が自らプログラムを書けたという実感を得ることにより，予習を継続する学習サイクルを作ることにある．

反転授業は，講義をｅラーニング化し，学生に自宅でその教材を閲覧させ，授業では実習中心の授業を展開する授業形式である[1]．授業を自宅で行い，自宅での宿題を大学で行うため，従来の授業方式と比べると学習の順番が反転する．LMS（Learning Management System）により，教員は学習履歴に基づく授業展開や学生支援が可能となる．

プログラミング教育は，講義で教員が言語の文法解説を行い，実習で学生が課題に取り組むのが一般的である．多田ら[2]による，全

Yasuhiro Hayashi*, Kenichi Fukamachi and Hiroshi Komatsugawa
Chitose Institute of Science and Technology
*E-mail: yasuhiro@photon.chitose.ac.jp

（受付：2013年10月５日，受理：2013年10月28日）

国12大学，専門学校におけるプログラミング科目のシラバス（合計89）の調査では，約72%の科目が従来の授業方式である．しかし，学生は講義で学習した内容を理解しても，実習でプログラムを書くと，講義で示されたサンプル以上のプログラムをなかなか書くことができない．中には，模範解答が示されるまで手を動かさない，友達が書いたプログラムを書き写すだけの学生もいる．このような状況では，学生はいつまでたっても自力でプログラムを書くことができないばかりか，実習時間を有効活用できていないことになる．学生の実質的なプログラミング能力向上のためには，従来のプログラミング教育の在り方を見直す必要がある．

本学では，情報通信技術を有する人材育成のために，情報系カリキュラムの編成とICT（Information Communication Technology）を活用した授業改善を図ってきた[3]．プログラミング科目として，１年次秋学期「情報技術概論」ではC言語の基礎（変数・配列，逐次，分岐，条件），２年次春学期「プログラミングスキル」ではC言語の発展（基礎の復習，関数，ポインタ，構造体），２年次秋学期「オブジェクト指向プログラミング」ではJava言語を開講している．履修学生の数は，１年次は３学科共通で行うため約240名，２年次は学科に配属されるため約80名である．

本学で作成しているｅラーニング教材は，教科書（図１）と演習問題から構成される．教科書には，解説文や図の他，アニメーションを用いている．これにより，プログラムの処理の流れと内容を対応付けて理解できるようにしている．演習問題では，問題の解答を穴埋め式または選択式で入力できる．このｅラーニング教材を用いたブレンド型授業により，板書や口頭では学生に伝わりにくい概念的な内容の解説を改善した．演習問題の機能を用いたWebテストにより，採点業務を効率化し，複数回の試験の実施を可能にした．また，FD（Faculty Development）の一環として，学部で教授する知識をまとめた知識集を整備し，ｅラーニング教材との関連付けにより，複数科目間で教材の共有化を図った．

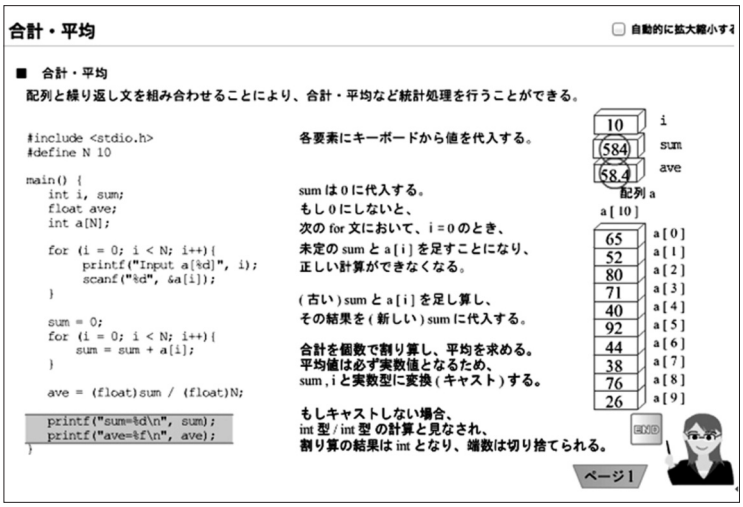

図1　ｅラーニング教材（教科書）の例

２．反転授業を取り入れたプログラミング教育の実践

本科目における授業内容・日程は，表１の通りである．これは例年と同じである．反転授業の導入に当たり，ガイダンスにて，①予習が必須である，②予習の仕方として，授業の１週間前からコース課題として提示される教科書を読み進め，内容が分かる部分と分からない部分を明確にすること．内容が分かる部分の演習問題を解くこと，③学習履歴が記録されること，を学生に周知した．学生は１年次からｅラーニングを利用しているため，その操作には支障がない．１～４週目では，「情報技術概論」にて学習したC言語の基礎の復習を予習とした．反転授業の導入という学習形式の変化により，学生が戸惑わないように授業設計を行った．５～13週目では，C言語で新たに学習する内容を予習とした．８週目では，授業前半の理解度を確認するために中間試験を実施した．14, 15週目では，最終課題を実施した．16週目では，学期全体の理解度を確認するために期末試験を実施した．

表１　授業内容・日程

週	授業内容	週	授業内容
1	ガイダンス，復習(UNIX基礎，ソフトウェア，変数の扱い)	9	ポインタ，中間試験の解説(希望者のみ)
2	条件処理	10	配列とポインタ
3	反復処理	11	関数とポインタ
4	配列	12	構造体（1）
5	関数（1）	13	構造体（2）
6	関数（2）	14	最終課題（1）
7	関数（3）	15	最終課題（2）
8	中間試験	16	期末試験

学習の流れを図２に示す．予習では，授業ポータルサイトとリマインダメールを通じて学生に予習を促し，学生は提示されるコース課題に取り組む．この間，TAに質問できる．講義では，前半に確認テストを行う．Webブラウザ上でその週の学習内容に関する簡単な問題を出題し，解答と答え合わせから，予習では学生が気付かなかった分からない箇所を明確にさせている．確認テストを実施している間，教員は学生の予習状況をLMS上で確認する．これにより，予習不足が疑われる学生に注意喚起が可能となる．次に，教員は予習にて用いたｅラーニング教材やスライドを使って，学生が分かりにくい概念的な内容やつまずき易い箇所を重点的に解説する．プログラミング言語の用語や構文など，理解し易い内容は，予習を前提とし，細かい説明は避けている．解説は，学生の集中力が切れないように15分以下を目安に行う．15分以上かかる内容は，解説を２回に分け，間に簡単な実習を挟むことにより，学生の手や頭を働かせるようにしている．また，聞き漏らしを防ぐために，解説中は学生にコンピュータの操作をさせない．これらの改善により，解説は昨年度より10分程度短くなった．実習では，基本課題と実習時間が増えた分，発展課題を行う．予習をし忘れた学生には，実習においてｅラーニング教材を自習させた後，課題に取り組ませている．基本課題では，例年と同様に文法の理解を深めるプログラミングを行う．発展課題では，じゃんケンゲーム（コンソール上で二人が互いの手を入力し，勝敗を判定する）のプログラミングに取り組む．その際，プログラムを書く上での注意点やアルゴリズムを記述したプリントを学生に順に配布し，サンプルより長いプログラムを書く過程を学習させている．基本課題は実習時間内に終わらせることとし，発展課題は宿題にしても良いとしている．

出席は，講義での授業ポータルサイトによる出席確認と実習での出席カードの受理が揃って，確定される．講義の冒頭に出席確認を行い，学生が講義の途中から参加して課題だけ行うことを防いでいる．また，学生が講義だけ参加して課題に取り組まないのを防ぐために，学生が基本課題を終えた時点で，TAがその課題を確認し，良ければ，出席カードを受理する．その後，学生には，発展課題に取り組むか，宿題にして教室から退出するか，選択させている．これにより，TAに課題を終えていない学生を重点的に指導させている．実習時間内に基本課題を終えることができない学生には，課題を翌週までの宿題とし，出席カードを配布する．後日，学生はTAに取り組んだ宿題を確認してもらい，出席カー

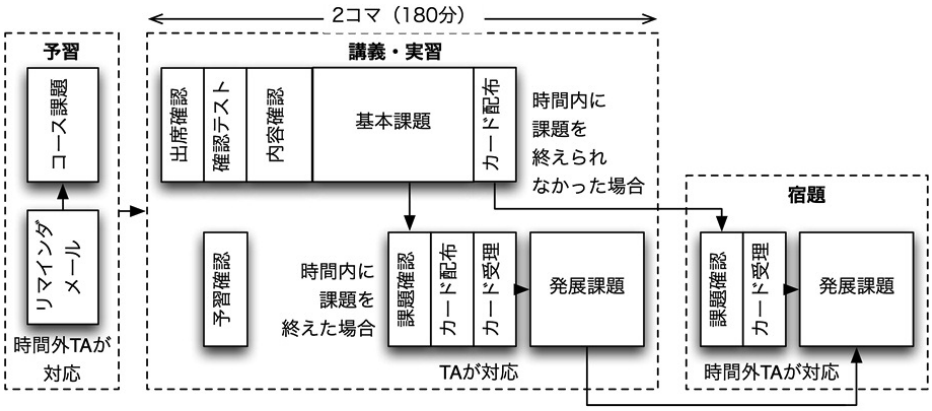

図２　「プログラミングスキル」での学習の流れ

ドを受理してもらう．その後，TAを通じて，学生に発展問題を取り組むように指導している．

中間と期末試験の実施方法を図３に示す．試験は知識を問うWebテストと技能を問う筆記試験から構成される．Webテストの結果は自動採点され，試験後，学生はその結果を確認できる．２種類の試験により，学生に知識と技能の差を把握させている．

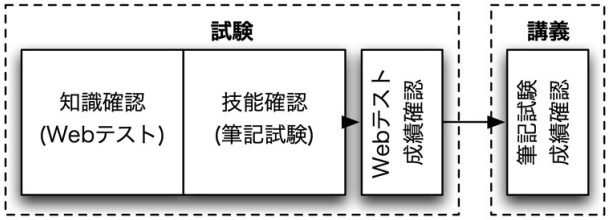

図３　中間および期末試験の実施手順

最終課題では，モグラ叩きゲーム（コンソール上で，モグラが無作為に出現し，座標を入力してモグラを叩く）を自作させている．学生に機能要件を提示して，その要件の達成度合いを評価する．さらに，学生に口頭試問を行い，自作したプログラムを説明できるかどうか確認している．

３．授業改善の効果および考察

昨年度と今年度の中間と期末試験の比較，学生アンケートと担当教員へのヒアリングから，本取り組みの効果を確認した．

中間と期末試験では，昨年度と同じ問題を用いた．表２,３に試験結果を示す．昨年度の学生数は57名である．中間試験は１～７週まで，期末試験は１～15週までが範囲である．昨年度に比べ，中間試験のWebテストと筆記試験の平均点と最低点が高く，標準偏差が小さかった．期末試験のWebテストでは，ずっと欠席していた学生が受験し，最低点が24となったが，出席学生だけの場合，最低点は48となった．期末の筆記試験では，昨年度は１問のみ実施した．今年度はこの問題を含めた8問を出題した．表３の今年度（2013）の欄に，昨年度と同じ１問だけの点数（右から２列目）と全体の点数（右から１列目）を示す．全体として理解度が昨年度よりも向上したと考えられる．

表２　昨年度と今年度の中間試験の結果

	Webテスト		筆記試験	
年度	2012	2013	2012	2013
最高点	100	100	97	98
最低点	24	48	0	21
平均点	79.3	86.9	66.3	70.1
標準偏差	14.6	11.6	18.2	17.7

表３　昨年度と今年度の期末試験の結果

	Webテスト		筆記試験		
年度	2012	2013	2012	2013	
最高点	98	98	4	4	95
最低点	7	24(48)	0	0	0
平均点	69.6	78	0.94	1.87	60.4
標準偏差	17.2	16.8	1.36	1.15	25.0

第16週に学生アンケート（有効回答数81）を行った．プログラムを書けるようになったか学生の実感度合いを調べた．「すごく書けるようになった」が７名，「書けるようになった」が58名，「あまり書けるようになっていない」が12名，「書けるようになっていない」が４名であった．さらに，反転授業をどう思ったか，かつ，実習を通じてプログラミングに面白さを感じたか，を調べた（図４）．

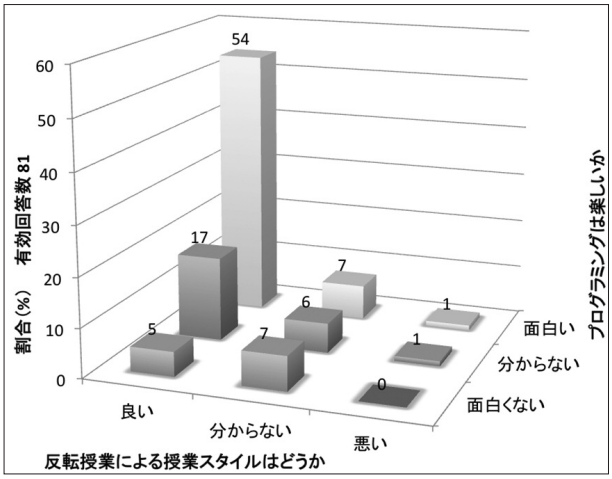

図4　反転授業とプログラミングの楽しさ

反転授業とプログラミングに対して，54%の学生が肯定的であった．また，「反転授業により授業内容をより理解できた」などの肯定

的なコメントを得た．一方で，反転授業の悪い点として，「予習を忘れると授業についていけなくなる」，「予習を行っていないと一人で取り組むため時間がかかる」などの否定的なコメントも得た．担当教員のコメントとして，例年に比べ，「学生がプログラムを書けている」，「難しいポインタを理解した学生が多い」を得た．

4．まとめと今後の課題

本学のプログラミング教育における反転授業の取り組みについて示した．昨年度に比べ，中間層以上の学生の成績向上に寄与できたと考えられる．今後，下位学生の早期発見とその対応のために，反転授業の授業運営の改善と e ラーニング教材の充実を図る．

参考文献

[1] Baker, J.W.: The ‘classroom flip’: Using web course management tools to become the guide by the side. Paper presented at the 11th International Conference on College Teaching and Learning, 2000.

[2] 多田 知正, 丸田 寛之: プログラミング教育における反復学習を採り入れた授業方式. 京都教育大学紀要, No.116, 2010.

[3] 林 康弘, 深町 賢一, 小松川 浩: ICT を活用した情報系科目における授業改善の取り組み. 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, pp.21–25, 2010.