Vol.2015-CE-128 No.21
2015/2/15

講義レスによるプログラミング実習教育の試み

河村一樹[†1]

The trial of the programming training education by lecture less

KAZUKI KAWAMURA[†1]

When I took charge of the subject "programming practice", the student's copy-and-paste problem was revealed. Then, this time, I tried the new classroom, instead of which does not do any lecture, the lecture was considered only as self-learning and individual guidance. The result, the copy-and-paste problem was lost, all of students programmed single-handedly and all of students, and most students approved how to advance this lesson.

1. はじめに

2013 年度から「プログラミング実習」という科目（半期，連続 2 コマ/週，2 単位，選択，1 年次から履修可）を担当することになった。教養的なプログラミング教育という位置づけのもとに,職業人としてのプログラマ育成ではなく，プログラミングを体験することによって問題解決力や論理的な思考力の育成を目指している。また，より実用的なプログラミング教育として 2 年次以降に「応用プログラミング実習」という科目を開講している。

2013 年度は，多くの大学や専門学校で取り入れらてる授業方式，つまり，講義（プログラミング言語の文法解説が中心）後すぐにプログラミング実習を行うという形式で実施した[1]。

しかし，期の後半から，落ちこぼれた学生によるプログラムのコピー&ペースト（以下，コピペと略す）問題が発覚した。このため，最終回において個別に面接試験を実施した際に，コピペについて調べた結果，半数弱の学生が部分的にでも経験したことが明らかになった。その理由について学生に聞いたところ，わからないことがわからないままになってしまうこと，授業のペースについていけないこと，提出が遅れることを恐れてついコピーしてしまうことなどが語られた。

この原因として，わからないことをわからないままにした結果，独力でプログラミングができなくなったことがあげられる。プログラミングは，自分でやってみることによって，構文の意味やロジックの組立てなどがわかってくるという特徴がある。このため，コピペに頼っていては，学習効果がゼロとなり，教員・学生ともに不幸である。

そこで，わからないことをわからないままにしないように，自分でやるという学習環境を生み出すために，講義は一切せずに自学自習で実習を進めるような授業を試みることにした。それとともに，授業中での個別指導を徹底することにした。

自学自習にあたっては，市販の教科書を講読した上で演習問題のプログラミングを行うこと，各演習問題についてのヒントを電子掲示板で参照できるとともにスレッド毎に個別に質問できること，共有ファイル上でプログラムの実行結果が確認できること，週単位の学習進捗を共有フォルダで公開すること, などについて指示した。また, e-Learning の LMS は使わないので，学習状況の把握のために「実習報告カード」の提出を義務づけた。

個別指導によって，学生個々人の実習の進捗度合いをチェックし遅れている場合はフォローアップを行い，学生のわからないことをわかるまで指導することができるようになった。

この結果，コピペ問題はなくなるとともに，学生全員が独力で演習問題のプログラミングを行ったことが明らかになった。

2. 2013 年度の授業実践

2.1 使用した言語

プログラミングのための言語は，スクリプト言語である JavaScript を用いた。JavaScript にした理由としては，次のようなことがあげられる[2][3]。

- テキストエデイタとブラウザさえあれば実行が可能（授業外での自学自習を想定しているため）
- Internet Explore にはスクリプトのデバッグ機能あり
- HTML に組み込むことで, より実用的なプログラミング（ウィンドウの動的制御，フォーム処理など）が可能（学生の動機付けに）

2.2 取り上げた内容

JavaScript の基礎的なプログラミングを行うものとし, 教科書は筆者がこの授業のために執筆したものを使った[4]。授業では, この中の「第 4 章 JavaScript」と「第 5 章 JavaScript

ⓒ 2015 Information Processing Society of Japan

の基本編」を主に取り上げた。なお,「第 6 章 JavaScript の応用編」と「第 7 章 JavaScript の実用編」については,「応用プログラミング実習」で取り上げることにした。第 5 章からは,節毎に演習問題を用意してある。これらは,その節で解説した内容（例題も含む）を理解すれば独力でプログラミングできるような一連の問題とした。

全 14 回の授業概要は,**表 1** のようになった。

表 1　全 14 回分の授業概要
Table 1　The lesson outline for a total of 14 times.

回数	取り上げた内容	演習問題(番号)
第 1 回	科目ガイダンス 1 : アルゴリズム，HTML と JavaScript	第 4 章の例題
第 2 回	科目ガイダンス 2 : プログラムの編集と実行，デバッグ方法	第 4 章の例題
第 3 回	画面への出力(document.write)	例題と 5-1-1
第 4 回	変数の扱い	5-2-1～5-2-2
第 5 回	代入文(=)，演算式(+=，-=，*=，/=，%=，++，--)	5-3-1～5-3-2
第 6 回	画面からの入力(prompt)	5-4-1～5-4-5
第 7 回	選択文(if)	5-5-1～5-5-2
第 8 回	選択文(if else)	5-5-3～5-5-4
第 9 回	選択文(switch)	5-5-5～5-5-7
第 10 回	繰返し文(for)	5-6-1～5-6-5
第 11 回	繰返し文(while)	5-6-6-～5-6-8
第 12 回	繰返し文(do while)	5-6-9～5-6-11
第 13 回	配列の扱い(new Array)	5-7-1～5-7-11
第 14 回	面接試験	－

ガイダンスでは，問題解決手順としてのアルゴリズムとはどういうものなのか，HTML と JavaScript の関係について取り上げた。

第 2 回目では, JavaScript のプログラム作成に使うエディタ（TeraPad を推奨，文字コードは UTF-8 で保存）の使い方，プログラムの実行に使うブラウザ（Internet Explore を推奨）の使い方，IE によるデバッグ機能の指定，ファイルの保存の仕方（各人のネットワークドライブか USB メモリ）について取り上げた。

第 3 回目以降は，節毎に取り上げたスクリプトの構文とそれに関連したアルゴリズムについて講義した上で，演習問題を各人でプログラミングするという実習を行った。その際に，授業時間内に指定した演習問題が提出できない場合は，次回の授業日までに宿題としてやってくるように指示した。

最終回では, 面接試験を実施した。学生個々人に対して, JavaScript の穴埋め問題を口頭で答えさせるとともに, 実習の進捗状況やコピペの有無などについて確認した。

2.3 コピペ問題

期の後半になると，落ちこぼれに見えた学生の中で，演習問題をコピペして提出するという事態が発覚した。それらの学生に直接事情聴取したところ，授業進捗が早くついていけなくなった，わからないことがわからないままになってしまった，独力でやろうとしてもわからないのでそのままとなってしまった，提出期限に間に合わせるため友人のソースコードをコピーして提出した，などの事情が明らかになった。

プログラミング実習という科目では，ソースコードが簡単にコピーできてしまうことや簡単なプログラムだとほぼ同じロジックになりコピーしてもわからないといったことから，コピペが横行する可能性が高い。

しかし，これでは教員も学生もお互いに不幸である。教員は学生にプログラミングの面白さを味わってもらいたいし，学生も自分の考えたプログラムが思った通りに動くことでプログラミングの面白さに気づくことになる。そこで，何らかの対策を講じることにした。

3. 2014 年度の授業実践

3.1 授業方式の改変

2.3 の問題には，わからないことがわからないままにも関わらず，授業が先に進んでしまい，提出期限に間に合わなくなるためコピペして提出してしまうことが原因としてあげられる。

そこで，わからないことをわからせるような仕組みを実現するために，講義ではなく自学自習を義務づけるとともに，授業中は個別指導を徹底するという授業方式を採用することにした。

講義を行うと，学習の進捗は学生ではなく教員に依存することになりやすい。教員は，シラバス通りの授業進捗を守るために先に進んでいく。その結果，授業についていけずに遅れた学生は，落ちこぼれたままとなり易い。それだけでなく, 一斉授業であるが故に, 教員にとっても学生個々人の理解度を把握することが難しくなる。これに対して，自学自習であれば, 学習の主体は教員ではなく学生になる。これによって，学生は自分の理解度合いに合わせて，学習を進めていくことになり，落ちこぼれたといった感はなくなる。

授業中の個別指導では，教員はチューターあるいはメンターとしての役割を担うことになる。学生のわからないことについては懇切丁寧に説明し，学生がわかるまで指導を行う。必要であれば，学生の持ち込んだプログラムを動かしながら，間違えの箇所を指摘することもある。また，進捗が遅れ気味の学生に対しては，自学自習の状況を把握してどのように改善すればよいのかのアドバイスをしたり，さらにはもう少し頑張ってごらんといった声がけをすることで心理的な支援を図ることもできる。これらによって，学生も徐々にプログラミングのおもしろさに気づくようになり，次の演習問題に取り組むモチベーションを高めることができるようになる。

ⓒ 2015 Information Processing Society of Japan

以上の授業方式は，反転授業[5][6]に近いパターンといえる。講義の部分は（予習としての）授業外における自学自習に置き換え，授業中は(アクティブラーニングとしての)個別指導だけに徹するという試みである（図 1)。ただし，反転授業の多くは，授業外の予習においては動画配信や e-Learning を用い，授業中に個別指導やグループワークなどの演習や実習を行うことを想定している。しかし，今回の場合は，動画や e-Learning を用いずに，教科書の講読とパソコンによるプログラミングの実習としている。このため，「反転的な授業」と言い換えることもできる。

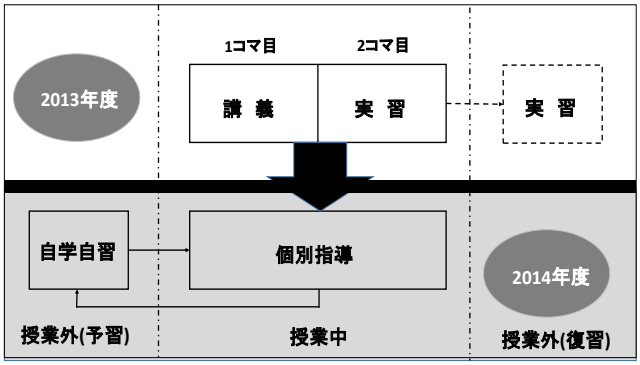


図 1　授業方式の改変
Figure 1　Change of a lesson system.

(1) 自学自習に関して

科目ガイダンスにおいて，講義をやらない代わりにすべて自学自習で行うこととした。具体的には，教科書の第 5 章の各節を自分で講読し理解すること，各演習問題に関してパソコンを使ってプログラミングすること，パソコンの利用については授業外では学内にある共用室か自宅やインターネットカフェでやること，などを指示した。また，自分のパソコンでもプログラミングの実習ができるように，環境設定（エディタのインストール，デバッガの機能，BBS の参照，共用フォルダの利用）を経験させた。

また，自学自習に合わせて，いくつかの支援策を講じた。

① 電子掲示板によるヒント集の公開

すべての演習問題について，ポイントとなる箇所についてのヒントを記載したスレッドを設置した。図 2 に，その表示の一部を掲載する。これによって，教科書で説明しきれなかった内容を網羅することができた。

なお，電子掲示板にしたのは，閲覧した内容についてわからないことがある場合に返信(Reply)をすることで簡単に質問ができること，それらの内容が他の学生も閲覧できることで参考になること，などがあげられる[7]。



図 2　ヒント集の画面例
Figure 2　The example of a screen of the collection of hints.

② プログラムの実行画面の公開

節毎の演習問題は，いずれも日本語によるプログラムの仕様だけを記述したものである。このため，そのプログラムが動作した結果どうなるかについてはっきりとしない場合がある。

そこで，すべての演習問題におけるプログラムの実行結果の画面を PDF ファイルとして共有フォルダにアップロードした。図 3 に，画面の例を示す。



図 3　演習問題 5-6-11 の実行結果例
Figure 3　The example of an execution result of the exercise problem 5-6-11.

ⓒ 2015 Information Processing Society of Japan

③　全学生の進捗状況の公開

ある演習問題を終えた段階で，そのプログラムの「ソースコードリスト」と入力したデータおよびプログラムの実行結果を表示した「画面リスト」そして「実習報告カード」一式を用意させ，それらを 1 週間分まとめて授業中に提出させた。それらをすべて確認して問題がなければ，演習問題実施済みとして受理した。このため，学生の進捗状況は 1 週間単位となった。

「実習報告カード」には，表 2 のような項目を用意し，すべて記入させて提出させた。これによって，e-Learning の LMS データと同様に，学習履歴を取得できるようにした。

表 2　実習報告カードの各項目

Table 2　Each item of a training report card.

区分	選択する項目
識別要素	学籍番号，名前，演習問題番号，提出日
実施場所	152 教室（授業で使った実習室） 大学共用室（1 号館 5 階） 大学構内（持ち込みノート PC で） 自宅・アパート・下宿先 各施設（カフェ，ファミレス，喫茶店など）
作業時間帯	授業中（月曜日の 4・5 コマ目） 授業外（休み時間，放課後）
実習時間	5 分以内 5 分以上 30 分以内 30 分以上 1 時間以内 1 時間以上
実施状況	独力でやった 先生に授業中聞いた 先輩に聞いた 友人に聞いた （正直に）コピペした 電子掲示板を参考にした

こうして，全員分の進捗状況をまとめ，それを共有フォルダ上で公開した（図 4）。これによって，誰でもが自分および他の学生の進捗状況を確認できるようにした。

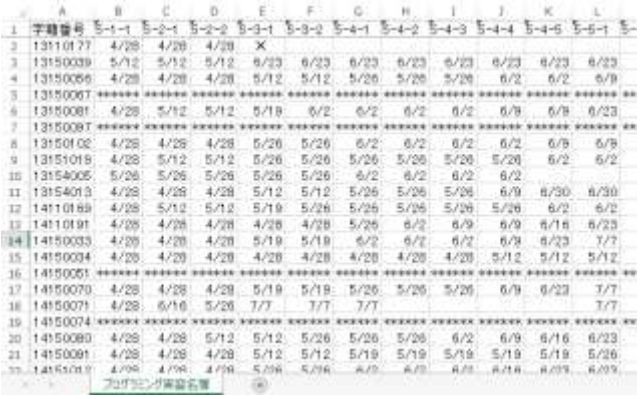


図 4　学生の進捗状況

Figure 4　Student's progress

(2) **個別指導に関して**

毎回授業開始時に，全員の進捗状況を示すことで，各人が自分の進捗を確認するとともに，他の学生の状況も把握することで遅れている学生に自覚を促すように指導した。

その上で，一人ずつ呼び出して，個々人の進捗を再度確認し，遅れている学生にはその理由を聞き出すとともに，頑張るようにとフォローアップを繰り返した。また，自学自習をしている中で，わからないことについての質問を受け，わかるまで徹底的にきめ細かく指導した。

この中で，最初の頃，最も多い質問が，単純なシンタックスエラーによるものであった。これには，キーワード等のスペルミスだけではなく，全角のスペース挿入も含まれた。仮名漢字変換を解除せずに，そのままスペースを打ってしまうことによって生じるエラーである。しかも，エディタでは，カーソルを 1 文字ずつ移動させて確認するしか手立てがなく，見つけ出すのにすごく手間がかかる。この結果，プログラミングの実習が先に進めないという事態に遭遇した。ただし，ある時期を過ぎると，学生も自分で対処できるようになった。

また，個別指導の間，待っている他の学生は，そのまま自学自習を教室内でも続けてよいとした。

4. 授業の実施結果と考察

2013 年度も 2014 年度も，前期 2 コマ/週で，いずれも 152 教室（50 台のパソコンを設置）で実施した。途中放棄を除外した履修者数は，2013 年度が 19 名，2014 年度が 14 名となった。

4.1 演習問題の提出状況

いずれの年度も演習問題 5-1-1 から 5-6-11 までを提出達成目標とした。これによって，JavaScript によるアルゴリズムの基本制御構造（順次・選択・繰返し）と基本的なデータ構造（データ型，配列）を習得することを目指した。なお，関数，探索と整列アルゴリズム，画面制御については，「応用プログラミング実習」で行うこととした。

年度別に全体の演習問題の提出状況をまとめると，表 3 のようになった。

表 3　全体の達成度合い

Table 3　The whole achievement degree.

年度	提出数	履修者数	学生一人当たりの提出数
2013	481	19	25.3
2014	394	14	28.1

これより，2014 年度の方が，学生一人当たりの演習問題の提出数が多く，それだけ多くの演習問題を解いて提出したことになった。

年度毎の演習問題の提出率（提出者数/履修者数）は，図 5 のようになった。

ⓒ 2015 Information Processing Society of Japan

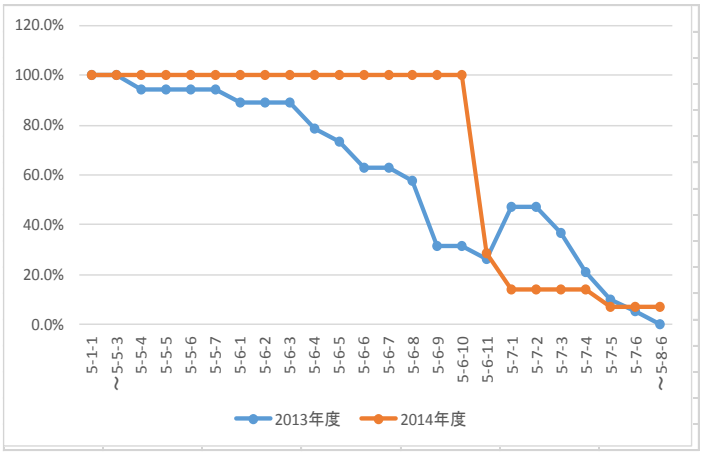


図 5　年度別演習問題の提出率

Figure 5　The presentation rate of the exercise problem by year.

これより，2013 年度は演習問題 5-5-4 から提出率が下がり始めたのに対して，2014 年度は演習問題 5-6-10 まで全員が提出した。これについては，個別指導において，毎回個々人の進捗を確認しながら遅れがち学生に対して積極的にフォローアップを図るとともに，わからないことを明らかにした上で自学自習に励むよう指導したことが要因といえる。一斉授業では，なかなか個人に対してきめ細かい指導ができないが，個別指導ではそれが可能になることで，このような提出率が達成できたといえる。

ただし，2014 年度において 5-7-1（配列に関する演習問題）あたりから進捗が遅れてきたことについては，学生の進捗状況を勘案し 5-6-11 までは提出するようにと指示したことによる。このため，遅れていた学生はここまでやるのが精一杯だったという結果といえる。

4.2　自学自習の状況

2014 年度に実施した「実習報告カード」をもとに，自学自習の状況について分析した。

(1)　自学自習した場所について

全体では，図 6 のようになった。

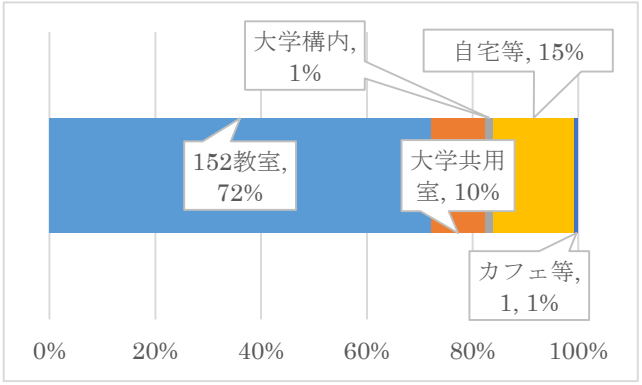


図 6　自学自習した場所（全体）

Figure 6　Place which carried out self-learning (whole).

このうちの 152 教室は，毎回の授業を実施したコンピュータ実習室であり，50 台のパソコンが設置されている。

当初，自学自習を行うほとんどの場所は，152 教室以外になると想定していたが，152 教室が 7 割強に達していた。これは，教員のアドバイスにより，授業中にわからないことが明らかになって実習を続けたこと，自宅にパソコンがない学生（2 名）や家族との共用のため使いづらい（7 名）がいたことによる。いずれにせよ，独力でプログラミングを行っていることがわかっていたので，教室での自学自習については自由とした。

また，図 6 のデータについて，授業日別にしたものが図 7 である。縦軸は，提出された演習問題の総数である。



図 7　授業日別（場所）

Figure 7　According to lesson day (place).

図 7 より，期の始めと終わりに，152 教室の利用が多かった。これについては，最初の段階では実習のやり方があまりよくわからないために 152 教室で実習を繰り返したこと，最後の段階については提出が迫ってきたため 152 教室でも実習を進めたことによると思われる。

(2)　自学実習した時間帯について

全体では，図 8 のようになった。

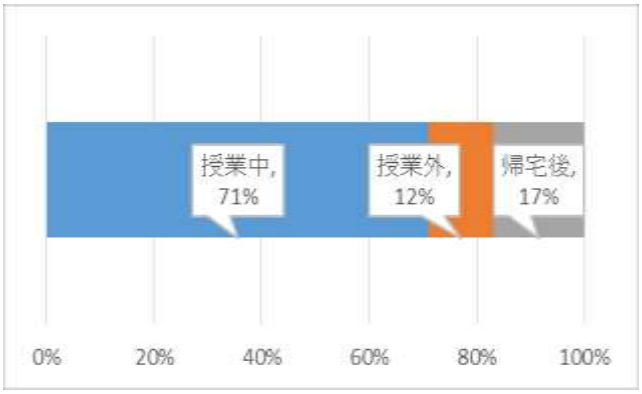


図 8　自学自習した時間帯（全体）

Figure 8　Time zone which carried out self-learning(whole).

ⓒ 2015 Information Processing Society of Japan

傾向的には，図 6 のほぼ同じ状況であった。これより，自学自習の約 7 割が，152 教室の授業中で行われたことになった。また，図 8 のデータについて，授業日別にしたものが図 9 である。縦軸は，図 7 と同様である。



図 9　授業日別（時間帯）
Figure 9　According to lesson day (time zone).

これについても，図 7 と同じような傾向となった。

以上のことから，来年度以降はできるだけ自学自習を実習用教室ではなく，かつ，授業外で行えるような手立て（e-Learning による自学自習の実施，LMS に蓄積される学習データによる学習状況の個別管理，授業外で起こる個人からの質問については電子掲示板の積極的な利用など）を別途取り入れることを検討したい。

4.3 アンケートによる分析

2014 年度の授業の最後に，履修者に対して記名式のアンケート調査（14 名全員が提出）を実施した。その結果を，表 4 にまとめる。

表 4　アンケート結果
Table 4　Questionnaire result.

NO	質問項目	回答項目	回答数
1	授業のやり方 1	いいと思う	11
		どちらでもない	2
		ある程度講義をしてほしい	1
		ちゃんと授業をしてほしい	0
2	授業の進め方 2	今のような個人面談だけでよい	10
		4 コマ目は講義で 5 コマ目は実習がよい	2
		4・5 コマ連続で講義と実習を混ぜながらやるのがよい	2
3	授業スタイルの有効性3	はい	10
		どちらともいえない	4
		いいえ	0
4	実習のやり方 4	授業中(152 教室)だけ	3
		授業中(152 教室)および放課後(共用パソコン室)	3
		授業中(152 教室)および帰宅後(自宅等)	2
		時々授業中(152 教室)おもには放課後(共用パソコン室)	2
		時々授業中(152 教室)おもに帰宅後	2
		すべて帰宅後(自宅等)	1
		その他	1
		すべて放課後(共用パソコン室)	0
5	わからない場合の対処5	授業中に先生に聞いた	12
		授業外に友人に聞いた	8
		電子掲示板のヒントを見た	8
		授業中に SA に聞いた	3
		インターネットで調べた	3
		共有フォルダにあるプログラムの実行結果を見た	2
		ほっていた	0
		その他	0
6	教科書について6	はい	3
		どちらでもない	2
		いいえ	9
7	演習問題の実施方法7	すべて独力でやった	7
		時々，友人に教えてもらった	5
		ほとんど友人に教えてもらった	2
		時々，友人のプログラムをコピペした	0
		ほとんど，友人のプログラムをコピペした	0
		全て友人のプログラムをコピペした	0
8	プログラミングスキル8	はい	9
		どちらでもない	4
		いいえ	1

質問の No.1 から No.3 までにおいて，自学自習と個別指導という授業のやり方については，大方の学生が賛成した。「ちゃんと授業をしてほしい」に対しては回答がゼロだったことから，学生も座学中心の講義だけを望んでいないことが明らかになった。座学による講義は，どうしても教員からの一方方向だけとなり，進捗管理も教員主導となる。これらの状況を打開できる授業スタイルとして，このようなやり方を取り入れることが妥当といえる。また，ほとんどの学生も，学習効果の有効性を認識していることがわかった。

No.4 の質問で，授業中だけで自学自習をした学生が 3 名いた。このうちの 1 名はパソコンを持っていない，2 名は家族との共用であった。自学自習を前提としたため，科目ガイダンス時に，パソコンを所有していない人は共用パソコン室でやるように指示した。しかし，授業外で時間をとって自学自習することに慣れていなかったことが考えられる。また，授業中でも自学自習を認めたため，教員だけでなく周りの友人などに直接聞けることから，この方が，実

1 「授業では講義を一切せずに，すべて自学自習するというやり方についてはどうですか？」
2 「授業の進め方についてはどうですか？」
3 「今回やった授業スタイルは，自分にとって有効ですか？」
4 「実習はおもにどうやっていますか？」
5 「自学自習していてわからない場合どうしましたか？」(複数回答可)
6 「教科書を読むだけで演習問題のプログラミングができましたか？」
7 「演習問題について，どうやっていましたか？(正直に答えてください)」
8 「この授業のやり方で，自分のプログラミングスキルが身につきましたか？」

ⓒ 2015 Information Processing Society of Japan

習がはかどったということも考えられる。

質問の No.5 から No.6 までにおいて，教員や友人にわからないことを聞くのが最も多かったが，電子掲示板のヒント集を参照している場合もある程度多かった。ヒント集では，すべての演習問題に関するヒントとなるであろうプログラミングのポイントについて解説している。このため，自学自習中にどうしてもわからない場合に利用する機会が多くなったと思われる。また，教科書だけでは，自学自習を進めることができにくいという意見も半数以上あった。

今回は，教科書以外に，電子掲示板によるヒント集の公開やプログラムの実行結果（画面表示）の公開を取り入れたが，今後は教科書の内容を e-Learning の教材コンテンツとして実装することを検討したい。そのときに，ヒント集および実行結果の画面についても，リンクを貼って参照できるような仕組みを考えたい。

No.7 の質問は，今回の教育実践において最も注視した内容である。この結果，演習問題はレベルの差はあるにしても，独力でプログラミングするとともに，コピペした学生は全くいないことが明らかになった。記名式のアンケートなので個人が特定できることから，回答の際に配慮した可能性もあるが，個別指導では独力でやることに意味があることを何度も強調し，学生も自分で作ったプログラムが思い通り動いたという達成感を体験したことで，このような結果になったといえる。

また，「ほとんど，友人に教えてもらう」と回答した学生が 2 名いたが，いずれも周りの友人に教えてもらう場面を何度も見ているので，きちんと回答したことがわかる。なお, このうちの 1 名は, パソコンは家族と共用であったが，演習問題の提出数が全履修者の中で 2 番目に多かった学生である。

No.8 の質問より，自学自習と個別指導による授業パターンによりプログラミングスキルが身についたとする学生が半数以上いた。これより，2014 年度の「プログラミング実習」の授業は，当初設定した教育目標をクリアできたといえる。また，学生にとっても，独力でプログラミングを繰り返したことで，スキルとして習得できたという実感を得ることができたともいえる。

5. おわりに

以上より，2013 年度に実施した旧来型の授業パターン（講義と実習）に比べて，2014 年度に実施した今回の授業パターン（自学自習と個別指導）の方が，学生個々人もコピペすることなく独力でプログラミングを行い，スキル定着の度合いも高く，学生もその有効性を認識したという結果を得ることができた。

今後は，反転的な授業をより完全なものにするために，e-Learning による自学自習の確立を目指したい。そもそも e-Learning は，自学自習に適した学習環境を提供しているといってよい。ネットが使える環境であれば，パソコンを使い，自分のペースに合わせて，あるいは，何度でも教材を繰返して学習することができる。また，わからないことがあれば，電子掲示板による FAQ や Q&A を参照したり，電子メールで直接問合せする機能もある。

それだけでなく，LMS により，個人の学習履歴データが常時収集されている。これによって，学習者の学習回数，学習時間，ドリルの正解率，テストの合否判定などの学習状況を個別に把握することができる。今回は，「実習報告カード」を提出することで代用したが，LMS での学習履歴の収集により，実際の学習経過および学習成果を詳細に把握できることから，個別指導がよりしやすくなるといえる。

さらに，プログラムの提出も授業時ではなく，できた時点で共有フォルダなどにアップロードする仕組みを考えたい。せっかく自学自習して演習問題をやり終えても，提出時期が最大 1 週間遅くなる可能性があり，このことが学生のモチベーションを低下させることも考えられる。

「プログラミング実習」は，本来スキル習得向けの科目であることから，その評価にルーブリック機能を利用することも検討したい。できれば，LMS の学習履歴を，何らかの形で e -ルーブリックに自動変換できる仕組みについても検討したい。現在のところ，多くの LMS では学習履歴データを利用者のニーズに合わせオーサリングする仕組みはない。これについても，今後研究を進めていきたい。

謝辞　日頃から反転授業の研究活動に関しての助言や支援を頂いている㈱ハンテンシャの加藤大氏に，謹んで感謝の意を表します。

参考文献

[1]　知正，丸田寛之：プログラミング教育における反復学習を取り入れた授業方式，京都教育大学紀要，No.116，pp.123-134，2010 年

[2]　河村一樹：情報ビジネス学科におけるプログラミング教育，東京国際大学論叢商学部編，第 84 号，pp.23-39，2011 年

[3]　河村一樹：一般情報教育におけるプログラミング教育のあり方について，情報処理学会研究報告，Vol.2011-CE-108(16)，2011 年

[4]　河村一樹：JavaScript による情報教育入門，大学教育出版，2011 年

[5] Bergmann, J., & Sams, A. : Flip your classroom: Reach every student in every class every day. Eugene, Or: International Society for Technology in Education,2012

[6]　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.677-683，2014 年

[7]　河村一樹：電子掲示板を用いたチュータリングの試みープログラミング教育での適用ー，e-Learning 教育研究第 7 巻，p.33-42，2012 年

ⓒ 2015 Information Processing Society of Japan