

写経型プログラミング学習と反転授業

‘SHAKYO’-Style Learning of Computer Programming and Flipped Classroom

京都大学 ○喜多 一 岡本雅子

Hajime Kita and Masako Okamoto

Kyoto University

Abstract In learning of computer programming by novices, typing example programs and executing them can be seen as a common practice. The authors paid attention to this practice by naming ‘Shakyo-style learning’ and discussed supporting methods to make this practice smooth. They also carried out practices of teaching computer programming asking learners to carry on this practice in parallel to learning of how programs work, and how to construct programs for particular tasks. Recently, learning outside class hours has been required more explicitly in teaching with background of quality assurance of teaching. ‘Flipped classroom’ attracts attention as a concept to improve such teaching. In courses of teaching computer programming with practice, ‘Shakyo-style learning’ within class hours has some difficulty due to differences in speed and accuracy of typing by learners. On the other hand, to carrying on it outside class hours is rather easier than to attack programming tasks independently. This paper discusses a course design of ‘flipped classroom’ type for programming which asks ‘Shakyo-style learning’ as an activity outside class hours.

1 はじめに

プログラミングの教育については現行の学習指導要領から中学校の技術科目で必修化され、また政府の産業競争力会議でも初等中等教育段階からの教育が検討されている [1]。一方、プログラミングの学習は初学者にとって様々な難しさを抱えており、数多くの研究が行われている [7] が実際の授業はなかなか改善されない。

プログラミングの学習において、例題のプログラムを自らタイピングして作成し、それを実行することはごく一般的に見られる過程であるが、プログラミングの教育において、必ずしもこれに十分に配慮した授業が行われてきた訳ではない。筆者らはこの過程を「写経型学習」と名付け、初学者がこれを実践する上での躓きなどに着目して支援方法を検討してきた [2, 3]。

近年、大学教育の質保証の点から大学設置基準で求められている授業時間外の学習を適切に行わせることが求められているが、その手法として「反転授業」が着目されている [4]。本授業では「写経型学習」という学習過程は自学自習に適することに着目し、これを授業時間外学習として課す形でのプログラミング教育の反転授業の構成法を提案し実践例を報告する。

2 写経型学習とその支援

写経型学習における躓きを防ぎ、促進するための具体的な支援策としては以下のようなものが挙げられ、著者

らが開発した教材 [3] は C 言語を例として文法的な事項の学習についてこの方針にそって編集されている。

エラーの明示的経験 教材や授業で敢えて明示的に典型的なエラーを経験させる。これにより、エラーを出すことが普通に生じることを理解させる。またエラーが生じた際の対処としてエラーメッセージを読むことを知るとともに、典型的な例でのエラーの原因を学ぶ。

音読、訳読への配慮 プログラミング言語では英数字以外にも類似の形状を持ったものなど多数の記号を用いる。初学者はこれらの記号を音読できず、そのため区別できないことが多い。また、プログラミング言語で現れる予約語等にも初学者には意味をつかみにくいものもあり、さらに C 言語では記号を多用した簡潔な表記を用いるため初学者には分かりにくい。

このような点に対して記号の音読やプログラムの日本語への／からの訳読などを取り入れ、「読める」ことを支援する。これにより、指導者とのコミュニケーションの失敗を防ぐとともに類似した記号の区別を促進することで、写経型学習でのタイプミスによるエラーの低減をはかることで後述の反転授業の実施を効率化する。

完全なプログラムと実行例の提示 プログラミングの指導書ではしばしば着目する機能に関連する部分のみのコードが記載されている。これに対し、初学者が写経型学習を実施するためには、実行可能な完全なプログラ

ム例が求められる。また、それが正しく動作していることを確認可能にするためには実行例も提示する必要がある。

また写経型学習の効果として、多数の例を読み書き実行することにより慣れを促進するという面がある。このために提示する例題は単に変数名を取りかえる場合のような単純でも慣れが求められるものは含め、他方で文法的な網羅性を示すための例は避け、実利用で現れるパターンを示すものをより多く選ぶ。

もちろん、写経型学習過程だけでプログラミングの能力が身につくとは言えない。そのため、これと並行してプログラミング言語の文法やプログラムの動作を理解すること、課題に対して構造的にプログラムを作成することを併せて学んで行くこともまた必要である。

3 写経型学習に着目した反転授業の構成

3.1 プログラミングの演習授業の問題点

プログラミングの授業は学習者が実際にプログラムを作成し動作を確認する演習を含む形式で行われることが多いが、プログラムを入力し実行する作業はタイピングの速度や正確性などに依存して学習者ごとの作業の進捗が相当にばらつく。このため、内容の説明とプログラミング作業を一齐に行う授業方式は、作業に時間を要する学生に配慮して実施せざるを得ないことから、全体の進捗が低くなってしまう。また、このような授業方式は解説や作業待ちなどの時間が多く、教員やティーチングアシスタントの能力を有効に活用しているとも言いがたい。

他方、大学での授業では大学設置基準に従えば、授業時間と同程度からその2倍の予習復習などの授業時間外学習が求められている。実態は、同基準の想定を満たしているとは言えないが、教育の質保証の観点から授業時間外学習が求められている。

プログラミングの授業では、授業時間外にプログラムの作成を課することが少なくない。しかしながら初学者にとって、発生したエラーに対応することも容易ではなく、また白紙の状態から独力でプログラムを作成することについては、どう取り組んで良いのかも見当がつかない場合さえある。学習レベルに不相应な課題を提示した場合、学習者に対して相当の時間的負荷を与えるだけでなく、学習意欲の低下を招くことも懸念される。

3.2 反転授業

近年、授業時間外の学習を促進する教授法として「反転授業 (flipped classroom)」が注目されている [4]。従来の講義型の授業では、授業時間内に一方的な知識伝達

を行うが、反転授業では、講義をビデオ教材などを用いた自習に置き換えることで知識の習得については学習者個人に任せ、討議など教員による指導を必要とする活動を授業時間内に実施する。こうした授業形態が取り入れられるようになった背景には、ビデオ教材やオンライン学習環境などを利用しやすくなったという情報インフラの拡充があるが、メディアの利活用にこだわらなくても、教員の指導を要しない学習内容については、授業時間外に実施し、授業時間内には教員の能力をより活用する指導を行うという反転授業のエッセンスは、様々な形で適用できるものと著者らは考える。

3.3 写経型学習による反転授業の構成

前節で述べたように写経型学習過程は例題を実際に入力して実行するという比較的単純な作業であり、作業時間は学習者によってばらつくが、適切な支援を行えば学習者自身で進めやすい。このことに着目するとプログラミング科目における反転授業の構成として以下のようなことが考えられる。

授業外学習 写経型学習過程を授業外学習として課す。

その際、エラーへの対処などは予め授業内で指導し、また適切な授業外学習の量を示すとともに進捗管理も行う。

授業内学習 最初の段階ではエラーの体験など写経型学習過程における躓きをなくす指導を行う。

それ以降は写経型学習過程と同期しつつ、単元の内容の解説、例題プログラムを改変して課題に取り組みさせるなど、教員の指導力を活かして学習内容の理解と定着をはかる活動を行う。

4 実践事例

4.1 実践対象

本報告では、京都大学で教養共通教育として全学部を対象に開講したコンピューターリテラシーの演習科目での実践事例を取り上げる。同授業は15週に渡って実施されるが、まず、学内ネットワーク、情報サービスや図書館の利用について講習した後、次に、データ処理、文書作成、プレゼンテーションなどのICT活用を演習形態で実施し、最後の4週程度をプログラミングの入門的な学習に充てる。なお、受講者は、文科系学部在籍者を主とする30名から40名の1年生であり、大半の学生はプログラミングの経験を持っていなかった。

4.2 授業計画

対象言語は前任者からの継続と教材 [3] の使用から C 言語を用いており、利用する処理系は学生所有の PC にも導入しやすい点から MinGW をコマンドプロンプトで操作する方法を選んだ。4 週程度の授業であるため教材 [3] は学生に貸与する方式を取っている。

授業は以下のように進めた。

- 前週の授業で教材を貸し出し、記号の音読などは予め教材を読む形で予習することを指示する。
- プログラミングのパート第 1 週
 - － 授業の目的、進め方の説明
 - － このパートの最終レポートとして課す課題の説明。理解のしやすい日常的に話題について計算機シミュレーションを通じて考察する課題を出している。
 - － プログラムの編集からコンパイル、実行までの手順の説明と演習。
 - － 教材に収められたエラーを体験する演習。
 - － プログラムにおける逐次実行と変数への代入について説明。
 - － 以降の写経型学習の自習課題（教材中の例題）と進捗をコース管理システム (CMS) で報告することの指示。
- 第 2 週
 - － CMS で報告された写経型学習の進捗を図示し、進捗を可視化する。
 - － 変数への代入と値の評価を再度説明と if 文による条件分岐についての解説。
 - － if 文の訳読演習。
 - － 計算機シミュレーション課題について、取り扱った内容を使う場面について解説、演習。
- 第 3 週
 - － 写経型学習の進捗の提示。
 - － for 文による繰り返し、配列の利用を解説。
 - － for 文についての訳読演習。¹
 - － 繰り返し処理を使った例題の演習としての Leibniz の公式を用いた円周率の計算。²

¹教材 [3] にはこれが含まれないため、授業用に作成。

²同公式は単純だが収束が遅い。このことを逆手にとって現代の計算機の速度を実感してもらう。

- － リダイレクトと CSV 形式の活用により結果を表計算ソフトで可視化する演習。
- － 最終課題のためのプログラムの骨格を示し、繰り返し処理を完成させ実行する演習。

● 第 4 週

- － 写経型学習の進捗を提示
- － 最終課題のための雛形プログラムを解説しながら、完成させてゆく演習。
- － 最終課題でのレポートに何が求められるかを説明し、残りの時間は課題に取り組む。

- 試験週：この科目では試験は行わないが、この週にレポート課題についての相談を受け付ける。

第 1～3 週では演習内容を早く終了した学生には写経型学習を実施させるが、併せて教えあいによる学習効果を期待して演習で躓いている学生の支援も要請する。

4.3 取り扱う内容

4 週の限られた学習機会であるためプログラミングに関する内容としては、逐次実行、変数（配列を含む）の利用、条件分岐、繰り返し、という基本的な事項だけを学ぶ。これらを用いることで課題についてシミュレーションを通じた考察を行うことを達成目標とした。関数については、C 言語の構成上、入出力をライブラリ呼び出しに頼るため、呼び出す形での利用を主に説明する。また文字列の処理は扱わず、ポインタについては関数呼び出しの引数としての利用に限定した。

課題については、学習者に一からコーディングさせるのではなく、雛形のコードを完成させてゆく形式で問題に取り組む過程自身を授業で体験する形を取った。

4.4 授業時間外学習の誘導

授業時間外に課す写経型学習については、学習者のペース作りが重要である。このため、期待される進捗を予め提示するとともに、授業の前日を締め切りとして CMS で進捗を報告させ、授業開始時にはクラス全員の進捗をグラフに示して他の学習者の状況と比較させた。

4.5 授業時間の利用

授業時には訳読教材など記入が必要なものは印刷して配布し、実際に記入させた上で指名して音読をしてもらう。課題の完成などを授業内演習として課すが作業内容を教員が口頭で説明することは冒頭だけに限定し、資料

を電子的に配布するとともに成果物は CMS で提出させることで、端末上で資料を見ながら学習者がそれぞれのペースで学習を進められるようにした。教員と TA は授業時間の大半を学習者の作業上の躓きへの対応や、理解の不完全な箇所の個別説明などに充てた。30~40 人もの受講者を対象とした進度の速い授業だったが、この 2 名で概ね対応が可能であった。

4.6 学習状況

授業時間外の写経型学習については、多くの学習者が課せられた量を実行した。また、最終課題についてもほぼすべての学生がシミュレーションプログラムを完成させた上で、シミュレーション結果についても一定の考察を加えて、レポートにまとめることが出来ていた。このように、学生の提出した提出物の評価から、授業では一定の成果を得たと見ることが出来る。ただし、授業終了後に学生から得た感想では、学習法として適切だったという意見が寄せられた一方で、ただ写して実行するだけという写経型学習に疑問を持つ意見も散見されている。

5 検討課題

本稿ではプログラミングの学習における写経型学習という過程に着目し、その実施における学習者の躓きを軽減する工夫について述べた。また本格的なプログラミングの授業ではないが写経型学習を予習として課す形で反転授業を構成した事例について述べた。授業実践を通して得られたいくつかの課題を述べておく。

学習者の動機づけ 本稿で述べた授業構成では写経型学習を予習的に行うものであるが、しばしば学習者の動機づけについて問われる。これについては次の点を分けて考える必要がある。

- プログラミングを学ぶことそのものへの動機づけ
- 授業時間外に写経型学習を行うことへの動機づけ

それぞれに一定の工夫は行っているが ARCS モデル [6] などを参照して分析・検討する必要がある。

プログラミング言語と処理系の選択 初学者に対するプログラミング教育でのプログラミング言語の選択は先の動機づけとも関連して重要な課題である。本稿で報告した例では C 言語を取り上げたが対象となる学習者にとって十分に適切であるという訳でもない。実践上の課題としても言語仕様上、コンパイル時のチェックが働きにくく実行時エラーが出やすいこと、コメントや文字列

に日本語コードを用いた場合、半角文字で書くべきところを全角文字で書いた場合に発見しづらいエラーとなることも少なくないことを挙げておく。

タイピングスキル 他国に比べてわが国の中等教育段階では PC の使用が少ないという調査結果もあり [5]、大学 1 年生では必ずしも PC の操作に十分習熟していない。近年、スマートフォンの急速な普及により日常的な PC の利用も減っており、キーボードのタイピングへの習熟度の低下が懸念される。

6 おわりに

本稿ではプログラミングにおける写経型学習過程に着目しその支援とこれを時間外学習として課す形での反転授業の構成について紹介した。写経型学習への着目はごく普通に行われている行為のフィールドでの気づきから生まれたものであるが、初学者のプログラミング学習においては、フィールドで暗黙知的に扱われているものもまだまだ多いと考えられ、フィールド情報学 [8] 的な視点で気づきを得て改善を考えることも重要である。

参考文献

- [1] 産業競争力会議：成長戦略の進化のための今後の検討方針 (2016/1/25)
- [2] 岡本雅子：模倣の重要性に着目した初学者向けプログラミング教育の研究，京都大学大学院情報学研究科博士学位論文 (2014)
- [3] 喜多，岡本，藤岡，吉川：写経型学習による C 言語プログラミングワークブック，共立出版 (2012)
- [4] 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)
- [5] OECD：Students, Computers and Learning, Making the Connection (2015)
- [6] J.M. Keller 著，鈴木訳：学習意欲をデザインするーARCS モデルによるインストラクショナルデザイン，北大路書房 (2010)
- [7] A. Pears et al: A survey of literature on the teaching of introductory programming, ITiCSE-WGR '07, 204-223 (2007)
- [8] 京都大学フィールド情報学研究会編：フィールド情報学入門ー自然観察，社会参加，イノベーションのための情報学一，共立出版 (2009)