

大学のプログラミング教育における反転授業と ルーブリックによる自己評価の導入

Introduction of Flipped Classroom and Self-evaluation by Rubrics to a Programming Course in Higher Education

盛 拓生^{*1}, 渡辺博芳^{*1,*2}, 水谷晃三^{*1}, 荒井正之^{*1}, 佐々木茂^{*1}, 古川文人^{*1,*2}, 高井久美子^{*1,*2}
Takuo MORI^{*1}, Hiroyoshi WATANABE^{*1,*2}, Kozo MIZUTANI^{*1}, Masayuki ARAI^{*1}, Sigeru SASAKI^{*1}
Fumihito, FURUKAWA^{*1,*2}, Kumiko TAKAI^{*1,*2}

^{*1} 帝京大学理工学部

^{*1} Faculty of Science and Engineering, Teikyo University

^{*2} 帝京大学ラーニングテクノロジー開発室

^{*2} Learning Technology Laboratory, Teikyo University

Email: takuo@ics.teikyo-u.ac.jp

あらまし: 大学におけるプログラミング教育の最初の科目において, 学生にプログラミングへの興味を持たせ, ゼロからプログラムが書ける力を身につけさせることを目標とする授業設計を行った. 目標を達成するために, 反転授業, 予習・授業・復習ワークシート, 上級生による少人数グループ指導, そしてプログラミング学習のためのルーブリックについて導入を検討した.

キーワード: プログラミング教育, 反転授業, 自己評価, ルーブリック

1. はじめに

本学情報電子工学科(旧ヒューマン情報システム学科)におけるこれまでのプログラミング教育カリキュラムでは, プログラミング科目の課題に対してもコピーアンドペーストによると思われるプログラムを提出する学生も少なくなく, そのため4年生の卒研配属の時点でも卒業研究を行うのに十分なプログラミング能力を身につけていない学生もいるような状況であった.

本学科では平成28年度入学者に対するカリキュラムの再設計を行ってきた. 本学科入学者に対する最初のプログラミング科目であるプログラミング1について, ゼロからプログラムが書ける, つまり, 与えられた問題に対してプログラムの内容を学生自身の考えとしてまとめてプログラミングできる力を身につけさせることを目標として授業設計を行った.

2. 授業設計

プログラミング1に対して, これまでのプログラミング教育の問題点を解決するための授業設計を行った.

2.1 プログラミング言語の選定

本学科ではプログラミング言語として近年はJava言語を用いたプログラミング教育を行ってきた. しかし, Java言語を用いてスタンドアロンで動作するプログラムを作成する場合, mainメソッド含むクラスを定義しなければプログラムを実行させることができない. このため初学者であっても, クラス, メソッドの概念を十分に学ばないうちに, プログラム中にclass, public static void main(String[] args)など理解できていない予約語を記述する必要があり, 本質

的なプログラミング学習に集中できない場合がある. このような状況を解決するためにいくつかのプログラミング言語について検討した結果, プログラミング1では, Processing⁽¹⁾をプログラミング言語として採用することとした. 理由は, 最小限の記述で実行可能なプログラムが書けること, グラフィックスなどの視覚的要素もプログラミングしやすいこと, Java言語との親和性が高いこと等による.

次にプログラミング1においてProcessingを教えるにあたって教科書を利用するかどうかについて検討した. 現在Processingに関するテキストはいくつか販売されているが, プログラミングの基礎的な概念を十分に網羅するものは見つからなかった. そのため, Casey Reas, Ben Fryらによるテキスト⁽²⁾の内容を中心とし, 不足する内容は独自に教材を用意することとした.

2.2 反転授業の検討

近年, 反転授業が注目されている. 本学でもプログラミング教育あるいは情報教育に対する実験的な導入, 検討がなされており, その有効性が確認されている^{(3),(4)}. プログラミング1においても, 次の方針のもと反転授業を導入することとした.

- インTRODクション, 試験以外の授業は原則的に反転授業とする.
- 反転授業では, 講義ビデオによる事前学習を義務付ける.
- 講義ビデオを視聴した後に, 学生自身に自身の理解度を把握させるための“事前ワークシート”への記入を義務付ける.
- 授業中に講義は行わず, グループ単位で学習活動を行う.

- グループへの直接的な指導は、2.4 節で述べる上級学部生が行う。

2.3 ワークシートの検討

これまでの本学科におけるプログラミング教育の問題点の1つに、課題のコピーアンドペーストがある。電子的にプログラミング課題の提出を行う場合に、コピーアンドペーストしたものを提出していると思われる例が少なくない。

プログラミング1では、学生にゼロからプログラムが書ける力を身に付けさせることを目標としている。これは、プログラムの内容を学生自身の考えとしてまとめてその構造を確認した上で、プログラミングすることを意味する。そのために、紙に印刷した課題をワークシートとして配布し、学生に手書きで解答を記入させ、その内容をスタッフが確認し、解答が正しければ計算機上で実行等を行うことを意味する。この場合、教員のみで全ての学生のワークシートをチェックするのは困難であると考え、2.4 節で述べる学生補助員を導入することとした。

ワークシートについては、各授業に対して反転授業における講義ビデオを授業の前に視聴していることを確認するための“事前課題ワークシート”、実際に授業内で取り組むための“授業内課題ワークシート”、授業で学んだ内容を振り返るための“振り返り課題ワークシート”を作成することとした。

ワークシート内の課題は、コーディング、コードリーディング、その他に分類し、それぞれ難易度のレベルを表示し、学生が自身の理解度に応じて課題を選択し、取り組むことができるようにした。

2.4 学生補助員によるグループ指導に関する検討

プログラミング1では反転授業の形式をとるため授業中の学生の活動の主な部分はワークシートに基づく課題の作成となる。その際に、学生は個々の課題について、紙で配布されるワークシート上に手書きで解答を作成し、学生補助員に解答をチェックしてもらい、解答が正しければ計算機上で実行等を行うこととなる。従って、実際の指導もワークシートを介して学生補助員が直接行うことが有効であると考えた。

その場合の学生補助員の指導方法、配置等については、森田らが実施している SA と呼ばれる学部学生補助員を利用した授業運営、演習教育の例⁶⁾を参考にすることとした。以降、学生補助員を SA(Student Assistant)と呼ぶ。

具体的には、1 回の授業を複数の教員で担当するチームティーチングで行うこととし、各教員は複数の SA からなる SA チームを担当する。SA チームを構成する個々の SA に対して 10 人以下の学生からなるグループを構成し、このグループ単位で学習活動を行うこととした。

2.5 ルーブリックの検討

2.3 節で述べたようにワークシートに取り組む際には、学生が自身の理解度に応じた課題を選択して

取り組むことができるようにした。しかしこれが有効に働くためには、学生の理解度、問題の難易度を学生が容易に判定できる仕組みが必要である。その仕組みとして我々はルーブリックを採用することとし、その設計を行った⁶⁾。紙面の都合上、その設計等の詳細については参考文献⁶⁾を参照していただきたい。

3. これまでの状況と今後の課題

これまで述べてきた授業設計の一部についてその有効性を確認するために、実際の開講前に SA 候補学生を対象とした Processing のプログラミング勉強会を実施した⁶⁾。その結果得られた知見に基づき授業設計を修正し、平成 28 年度 4 月よりプログラミング1を開講した。履修者は留学生等を含めて 60 名程度であり、これを担当する SA は 10 人である。この状況で 8 人程度の学生を 1 人の SA が指導している。事前学習については事前課題ワークシートにより行っている。5 回目までについては、出席者が 55 名前後で、その 9 割程度が事前学習を済ませている。

既知の問題点としては、事前学習をしてこない学生、欠席が多い学生への対応、学生へのルーブリックの理解、記述、利用の周知、SA に対する指導、指示の徹底、振り返り課題シートの効率的な配布、回収の手段などが挙げられる。

4. おわりに

本稿では本学科の新入生にとっての最初のプログラミング科目であるプログラミング1の授業設計について述べた。学生にプログラムへの興味を持たせ、プログラムをゼロから書ける力を持たせるために、反転授業、各種ワークシートの導入、SA による少人数グループ学習、学生の自己評価のためのルーブリックの導入について検討した。現在この授業計画に基づき実際に授業を行っている。その結果についてはまた改めて報告したい。

参考文献

- (1) Processing ,<https://processing.org>, (参照 2016.06.06).
- (2) Casey Reas, Ben Fry 著, 船田 巧訳: “Processing をはじめよう (Make: Projects)”, オライリージャパン(2011)
- (3) 水谷晃三, 高井久美子: “プログラミング初学者を対象にした動画教材による反転授業の実践と評価”, 情報処理学会研究報 Vol.2015-CLE-17, No. 34, pp. 1-8 (2015)
- (4) 渡辺博芳, 高井久美子: “「情報基礎」におけるビデオ講義を用いた反転授業の評価”, 情報処理学会論文誌教育とコンピュータ Vol.1, No.4, pp.64-74 (2015)
- (5) 森田 彦: “SA を活用した授業運営: プログラミング演習の場合”, 札幌学院大学社会情報第 18 巻 2 号 pp. 117-129 (2009), <http://hdl.handle.net/10742/1> (参照 2016.06.02)
- (6) 渡辺博芳, 水谷晃三, 盛 拓生, 荒井正之, 佐々木茂, 古川文人, 高井久美子: “大学のプログラミング教育のためのルーブリックの検討”, 情報処理学会研究報告, Vol.2016-CLE-19, No. 6, pp. 1-9 (2016)