

大学初年次を対象としたプログラミング入門科目の展開

Introductory Programming Courses targeted at university first year

坂田 圭司*1

Keiji SAKATA*1

*1 東海大学情報教育センター

*1 ICT Education Center, Tokai University

Email: kgsakata@tokai-u.jp

あらまし：東海大学情報教育センターでは、2014年度から「プログラミング入門」科目を設置して、学生の所属学科や専攻を問わないプログラミングスキル教育を開始した。授業は、Scratch を用いたプログラム要素と処理の流れの理解から始め、Processing によるグラフィックスプログラミング、計測制御プログラミングを経て、グループワークによる作品製作までを内容としている。今回の発表では、実践結果に関して報告を行う。

キーワード：プログラミング、CS アンプラグド、Scratch, processing

1. はじめに

大学初年次において、プログラミング学習を実施することは、ソフトウェア開発スキルを向上させるだけでなく、試行錯誤による問題解決力を身に付けられることから、専門分野を問わず幅広い学科や専攻の授業で取り入れるべき内容である。

従来の大学での情報教育では、パーソナルコンピュータの操作と利用を中心とした授業が多く開講されてきた。情報端末の機種とソフトウェアが多様化した近年において、授業内容の刷新が課題となっている。学生は卒業後に高度に情報化されつつある社会での活躍が期待されている。これらの現状で、大学での情報教育は、高度な ICT の活用による問題解決力を身に付けることに主眼を置いたカリキュラムを展開する必要がある。

東海大学情報教育センターでは、湘南キャンパスの学科・専攻に所属する学生を対象に、情報の授業を開講して年間延べ1万人以上が履修している。

これらの学生のうち、主に初年次を対象として、2014年度春学期から「プログラミング入門」科目を新設して、学生の問題解決力を向上させることを目的に、言語に依存しないプログラミングスキルの習得を行う授業を展開している。

2. 初年次でのプログラミング学習の必要性

大学でのプログラミング授業は、理工系の専攻で多く設置されてきた。授業では、限られた授業時間を考慮して、基本的な文法と構文の理解を中心とした内容が多い。このため、アルゴリズムの理解が不十分のまま進み、実験や卒業研究等でプログラミングスキルの必要性が高まった段階で、再び学び直す学生が多く見られる⁽¹⁾。プログラミング入門の履修者28人に対して、授業初回に実施したアンケートでは、10人が大学の授業でのプログラミング経験があるにも関わらず、学び直すことを求めて履修したことが判明した(表1)。

表1 授業でのプログラミング経験 (回答数 28)

	回答	人数
1	習ったことがない	16
2	中学校以前から経験あり	0
3	高校で経験あり	2
4	大学で経験あり	10

また、同アンケートでのプログラミングに対するイメージでは、「とても難しい」が12人、「結構がんばれば出来そう」が9人で、全体の75%が苦手意識を持っていることが分かる(表2)。

表2 プログラミングへのイメージ (回答数 28)

	回答	人数
1	とても難しい	12
2	結構がんばれば出来そう	9
3	少しがんばれば出来そう	6
4	普通に出来る	0
5	得意な方だと思う	1

上記のアンケート結果から、大学初年次ではプログラミング経験者が非常に少なく、プログラミングスキルを身に付けるための入門授業の必要性が確認できた。

3. プログラミング入門の内容

プログラミング学習は、他の授業では身に付けにくい下記の能力向上に効果的である。

- (1) 目標を設定する力
- (2) 必要な項目を考える力
- (3) 処理の手順を考える力
- (4) 誤りを見つけて解決する力

特にプログラムのバグとその修正は、「失敗から逃げるのではなく、原因を見つけて問題解決する」ことを、実習を通して受講者に実感させる大きな役割

を果たせる。これらは、学生の専攻分野を問わず、高学年での授業や就職後の社会活動でも効果があると考えられる。

本センターのプログラミング授業は、言語毎に開講していたプログラミング科目を整理して、プログラミング入門・基礎・応用の3段階で系統的な学習が行えるようにカリキュラム改定を実施した。

プログラミング入門は、初年次のプログラミング未経験者を主な対象として、CS アンブラグドと学習ツールを用いた授業内容としている。2014 年度春学期開講授業で、本研究の対象となった授業計画を表 3 に示す。

表 3 プログラミング入門の授業計画

授業内容	習得目標
1: ガイダンス	概要, 学習環境の把握
2: CS アンブラグド学習 1	コンピュータに依存しない
3: CS アンブラグド学習 2	問題解決の考え方
4: Scratch 実習 1	プログラムの要素・処理の流れ, オブジェクト指向の基本
5: Scratch 実習 2	
6: Scratch 実習 3	
7: 計測制御プログラミング 1	入出力の実践
8: 計測制御プログラミング 2	
9: Processing 実習 1	グラフィックス
10: Processing 実習 2	ユーザーインターフェース
11: Processing 実習 3	
12: グループ作品製作 1	グループワークによる実践
13: グループ作品製作 2	
14: グループ作品製作 3	
15: 作品発表, まとめ	
16: 定期試験	知識の確認

授業目標は、「自分でプログラムを作り出せる基礎力を身に付ける」ことであり、プログラミング言語の文法や開発環境の操作を習得することは中心では無い。

授業の前半は個々でのスキルアップを主として、後半はこれまでの学習内容を組み合わせたプログラム作品を4人一組の協同で製作することで、実践的な企画・設計・作成・検証の流れを体験できるようにしている。

学習ツールは、(1) 学習内容に適しているか、(2) 操作が煩雑でないか、(3) 利用環境を選ばず学生の予習復習が容易か、という条件を満たすソフトウェアを選択をした。

Scratch は、MIT が開発したプログラミング学習ツールで、世界中の初等中等教育では多く使われている。大学初年次においても、最初の段階においては、プログラミングのエッセンスを習得するには効果的なツールである。

これまでの調査では Scratch での学習後に、高水準

言語によるプログラミングへステップアップすると、操作や内容のギャップが大きく、学生の学習意欲と理解度が低くなる傾向が見られた⁽²⁾⁽³⁾。プログラミング基礎の授業は、言語によるコーディングが必須であり、学習内容の連続性を持たせる必要がある。

これを解決するために、Processing によるプログラミングを次の内容とした。Processing は、シンプルな文法で充実したグラフィックスプログラミングを行える特徴を持つ。Java 言語をベースとしているため、次のステップでの他のプログラミング言語への導入となる。また、開発環境はシンプルで習得に余計な時間を必要としないため、入門段階での授業利用に向いている。Scratch と Processing の現行バージョンは、Web アプリで提供されており、教室環境への依存を最小限に授業活用が可能である⁽⁴⁾⁽⁵⁾。

さらに、Arduino、計測制御プログラマー、ロボットカーを用いた計測制御プログラミングを学習内容に加え、コンピュータにおける多様な入出力を扱うスキルを身に付けることを目指している。

4. まとめ

日本では大学入学時においてもプログラミングの経験者が少なく、従来の授業内容では学生の学習意欲や理解度向上させ、高学年での実践に繋げるのは難しい。

初年次において、言語に依存しないプログラミング学習を実施することは、分野を問わない問題解決力の向上に繋がることを期待できる。

本研究では、新たに設置したプログラミング入門科目で、プログラミング言語に依存しない学習を実践して、学生の意識変化やスキル向上を調査している。発表では実際の授業内容を報告するとともに、2014 年春学期における成果報告を行う。

参考文献

- (1) 坂田圭司, 「大学における情報教育の現状と展望」, 日本情報科教育学会第 6 回全国大会 (2013)
- (2) 坂田圭司, 遠藤陵二, 「高大連携におけるプログラミング学習の実践」, 日本情報科教育学会第 4 回全国大会 (2011)
- (3) 坂田圭司, 遠藤陵二, 「ソフトウェア開発における基礎学習の実践と評価」, 日本情報科教育学会第 5 回全国大会 (2012)
- (4) 坂田圭司, 高橋隆男, 「データ可視化および組み込みプログラミングを用いたソフトウェア開発学習の試み」, 平成 25 年度 ICT 利用による教育改善研究発表会 (2013)
- (5) Cassy Reas, Ben Fry 著, 船田巧訳, 「Processing を始めよう」, オライリー・ジャパン (2011)