

# C言語プログラミング教育における反転授業の実践

小島篤博

大阪府立大学 大学院 人間社会システム科学研究科

## Practice and Evaluation of Flipped Classroom for Teaching C Programming Language

Atsuhiko Kojima

Graduate School of Humanities and Sustainable System Sciences,  
Osaka Prefecture University

Introducing a style of flipped classroom into a course of learning programming bring the benefits that students can make more exercise in the class time by learning basic syntax and algorithms in advance. In this paper, a practice of flipped style education on a programming course for the first-year students is reported including short movies for preparation and assurance tests. An analysis and evaluation on the practice are also presented.

キーワード: プログラミング言語教育, 反転授業

## 1 はじめに

大学における情報系のプログラミング教育においては、コンピュータを用いた基礎的なアルゴリズムと、プログラミング言語の文法を修得することを目的としている。しかしながら、一般にプログラミング技能の修得には、自分でプログラムを作成、コンパイルし、動作を確認するという一連の演習が不可欠であり、通常の授業内では時間が不足するという慢性的な問題を抱えている。多くの授業では、プログラミングの考え方や文法を教師が時間内に講義し、課題については個々の学生が時間外に取り組むといった形態が取られている。

一方、近年アクティブ・ラーニングの一手法として、授業と時間外の自習との役割を逆転した反転授業の導入が試みられている<sup>(1)(2)</sup>。反転授業の長所としては、事前に学習内容をビデオ等の自習教材で予習することで、授業では演習を中心とした能動的な活動に多くの時間を割り当てることができるという点が挙げられる。この利点は、前述のように演習の比重が大きいプログラミング教育においても効果的であると考えられるが、大学のプログラミング科目における実践の報告例はまだ多くない<sup>(3)</sup>。

本論文では、大阪府立大学の情報系課程（現代システ

ム科学域 知識情報システム学類）におけるプログラミング科目として、1年次後期に担当されている「プログラミング入門」において、反転授業用のビデオ教材や予習確認のための小テスト等を開発し実践を行なった事例について報告する。

## 2 プログラミング科目の概要

まず、授業実践の対象とした科目「プログラミング入門」の授業目標および授業計画について説明する。

### 2.1 授業目標

授業目標としては、問題解決のための手法として基礎的なアルゴリズムを理解し、C言語の基本的な文法とプログラムの作成方法を習得することである。ただし、科目を配当している課程は情報系ではあるが工学系ではなく、将来的にはプログラム開発技術者と言うより、広くシステム構築の企画・運営に携わる職種を目指す学生が多い。また、研究室配属後に使用するプログラミング言語も、Java、Swift、Python、Rなど研究室によって大きく異なっている。このため、到達目標としてはC言語の基本的な文法はカバーしつつも、言語によって仕様

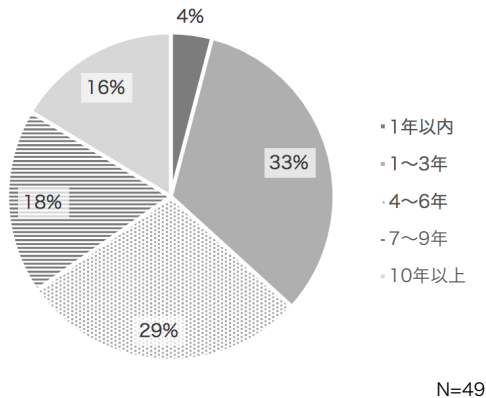


図 1: パソコン使用経験

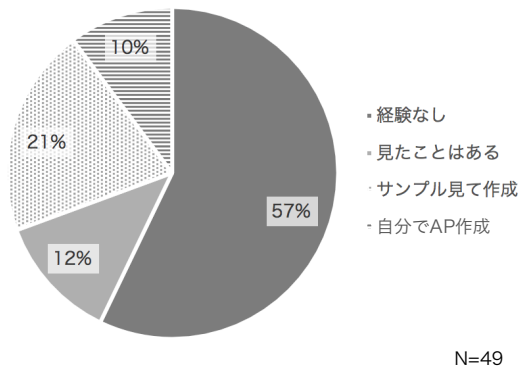


図 2: プログラミング経験

の差異が大きい構造体やポインタは授業内容から外している。

## 2.2 授業計画

授業計画の参考とするため、授業開始時に受講者のプロフィールを把握するためのアンケートを実施した。図 1 は、これまでの PC 使用経験を尋ねたものであり、最も多いのが 1~3 年、さらにほとんどの学生がそれ以上の経験を持っていることがわかる。本学では 1 年次前期に共通教育として情報リテラシー教育を実施しており、PC 未経験者はいない<sup>(4)</sup>。また、図 2 はプログラミング経験について尋ねたものであり、約 7 割の学生が自分でプログラムを作ったことはないと回答している。プログラミング経験については年度によって変動はあるものの、大半の学生が未経験であり、授業としては初めてプログラミングを行う初学者であることを想定する必要がある。

表 1: 「プログラミング入門」の授業計画

項目	内容
1 C 言語とは	C 言語の概要 プログラムの作成手順
2 テキストの出力	文字列出力, printf
3 変数とデータ型	変数名, 変数宣言, 基本的なデータ型
4 配列	配列の構造, 配列の宣言
5 計算と演算子	演算子と数式, 変数の出力
6 データの入出力	数値・文字列の入出力, scanf, fgets, getchar
7 条件分岐 (1)	if 文, 関係・等価演算子
8 中間テスト 分岐処理 (2)	1~7 回の内容 switch 文
9 反復処理 (1)	定型的な反復, for 文 制御変数, マクロ
10 反復処理 (2)	汎用的な反復, while 文, break/continue 文
11 反復処理 (3)	反復処理の応用, 素数判定, 組合せ探索
12 関数の定義	関数定義の文法, プロトタイプ
13 さまざまな関数	標準ライブラリ, #include strlen, isalpha, sin, cos
14 ファイル処理	ファイル処理の手順, fopen, fgets, fclose
15 期末テスト ソート	8~14 回の内容 バブルソート

これらに基づいて、初めて C 言語を学ぶ学生が一通りの文法を習得し、プログラムが作成できることを目標とした授業計画を表 1 に示す。教科書は林晴比古『明快入門 C スーパービギナー編』<sup>(5)</sup>を採用した。

各回の授業は基本的に次のような手順で進める。

### 1. 予習ビデオの視聴

授業日の 1 週間程度前から、その回の内容を予習するためのビデオを公開する。受講者には授業開始前までに視聴することを指示する。

### 2. 予習確認テスト

授業の最初に、予習ビデオの内容を確認する Web 上の小テストを実施する。出題は 2~4 問程度で、問題プールからランダムに出題する。得点は成績にも反映する。

### 3. プログラム作成演習

その回の学習内容に対応したプログラムの作成を演習形式で行う。受講者は教師に質問したり、受講者同士で相談したりしながら進める。

### 4. 課題提出

指定された課題を 1 週間後までに提出させる。課

題は授業中に提示するようにし、少し取り組む時間を与える。一旦提出した課題で間違いなどがあれば指摘し、再提出させる。

なお、第1回は初回であるため、授業の進め方についてガイダンスを行い、予習確認テストは実施しない。中間・期末テストは授業の一部として実施し、テスト終了後は通常の授業を続けるものとした。

### 3 反転授業教材の開発

前章で説明したプログラミング科目を反転授業で実施するための教材として、予習ビデオと確認テストを開発した。大阪府立大学では、すべての開講科目で利用可能な公式サービスとしてオープンソースのLMSであるMoodleを運用しており、これらの教材もMoodle上で提供している<sup>(6)(7)(8)</sup>。

#### 3.1 予習ビデオ

今回予習ビデオとして作成したものの一覧を表2に示す。形式はA,B,Cの3種類からなり、Aは教師によるスライド説明を撮影したもの、Bは教師がその回の概略を口頭で説明したもの、CはPowerPointのスライドに音声のみを吹き込んだものである。それぞれ再生時間があり長くなり過ぎないように、6~8分を目安として作成したが、一部10分を超えたものもある。図3に予習ビデオの例を示す。

反転授業では、受講者は毎回授業前にビデオを視聴し理解しておくことが求められるが、一般にスライドに音声吹き込んだもののみでは単調となってしまうがちであり、飽きさせずに視聴を促す工夫が必要である。本研究では、香川大学の林が実践している事例に倣い<sup>(9)</sup>、予習ビデオの本編であるスライド形式のビデオに先立ち、教師がその回の概略を説明するという態で毎回異なる場所を訪問し収録を行なった。

スライド形式のビデオは、前年度以前の通常授業で使用していたPowerPoint形式の資料に音声を収録し、ビデオ出力したものを動画編集して作成した。PowerPointへの収録は、説明用の原稿をあらかじめ用意し、スライドごとに音声を吹き込むという手順で行なった。当初は原稿を用意しない形で収録を試みたが、何度もリテイク

表 2: 予習ビデオ一覧

	形式	内容
1	02:58 A	ガイダンス, 導入
	13:11 C	C言語の概要, コンパイル
2	08:40 C	文字列の表示, main関数
	06:32 C	基本的な文法
3	01:10 B	授業の概略
	08:03 C	変数, データ型
	06:12 C	代入, 変数の出力
4	01:23 B	授業の概略
	06:06 C	配列, 文字列
	05:51 C	初期化, 定数
5	01:37 B	授業の概略
	07:57 C	算術演算子, 優先順位
	07:21 C	型変換, インクリメント
6	01:14 B	授業の概略
	06:18 C	実数の入出力, scanf
	05:39 C	文字列の入出力
7	01:18 B	授業の概略
	06:46 C	制御構造, 分岐処理, if文
	08:26 C	多分岐, 条件式
8	01:23 B	授業の概略
	08:52 C	複合条件, switch文
9	01:50 B	授業の概略
	07:30 C	反復処理, for文
	04:04 C	総和計算
10	01:22 B	授業の概略
	07:10 C	多重ループ
	07:24 C	while文, break/continue文
11	01:28 B	授業の概略
	09:17 C	字下げ, 素数判定
	03:06 C	組合せ探索
12	01:24 B	授業の概略
	08:18 C	関数定義, ローカル変数
	08:23 C	return文, プロトタイプ
13	03:40 C	授業の概略, グローバル変数
	09:24 C	標準ライブラリ関数
14	10:22 C	授業の概略, ファイル入出力
	04:09 C	コマンドライン引数
15	01:56 B	授業の概略
	08:52 C	アルゴリズム
	03:50 C	乱数, バブルソート

することになり結果として収録時間が長引いてしまうという経験に基づいている。

#### 3.2 確認テスト

毎回の授業で予習ビデオの内容を確認するための小テストを作成した。授業のガイダンスを行なった初回、および中間・期末テストを実施した2回分を除く12回分であり、内容は予習ビデオで説明した中から出題している。問題はMoodleで作成可能な多肢選択、短文記述、穴埋め、組合せ、正誤の中から問題ごとに選択し、それぞれ類似問題を3~5問程度用意してランダムに出題できるようにしている。図4に出題例(第5回)を示す。



### for文の構造

KIS

・変数の値と連動しながら、決められた回数の反復処理を行う。変数が反復回数を制御することから、**制御変数 (カウンタ)** と呼ばれる。

最初に一度だけ、制御変数 i の値を設定する

毎回の処理の前に、継続条件を判定する  
→ 偽なら終了

毎回の処理の後に制御変数を更新する

```
for (i = 1; i <= 10; i++) {
    printf("Hello\n");
}
```

図 3: 予習ビデオの例

また、予習確認テストとは別に、第 8 回と 15 回にそれぞれ中間・期末テストを作成した。これは、前半と後半の内容の知識定着度を判定するために実施するものであり、それぞれの範囲から出題している。問題自体は予習確認テストと同様の形式であるが、重複はしていない。出題数は、予習確認テスト 33 問、中間・期末テスト 13 問、ランダム出題のために用意した問題の総数は 206 問である。

## 4 授業実践と評価

以上の教材を利用し、授業での実践を行なった。対象科目は、大阪府立大学の現代システム科学域 知識情報システム学類 1 年次後期配当の「プログラミング入門」であり、毎週金曜 5 コマ (16:15~17:45)、期間は 2016 年 9 月 30 日~2017 年 2 月 3 日の計 15 回である。科目の受講者数は 53 名、このうち学習履歴の研究利用に同意した 49 名を対象に、以下の分析を行なっている。

### 4.1 ビデオの視聴行動

まず、予習ビデオの視聴行動について、Moodle のログから統計を取った。図 5 は、各回のビデオ視聴率 (受講者数に対するユニーク視聴者数の割合) の推移を示す。当初、特に 1~3 回目までは 8 割以上の受講者が視聴した上で授業を受講していたが、4 回目以降はほぼ 6

次の数式を C 言語の文として記述せよ。(変数 x, y は宣言されているものとする)

$$y = x^2 - 3x + 1$$

解答:

次の数式を C 言語の文として記述せよ。(変数 x, y は宣言されているものとする)

$$y = \frac{1}{x-3}$$

解答:

次の文を実行した後の、変数 i, j の値を答えよ。

```
int i, j;
i = 3;
j = -1;
```

i の値

j の値

次の数式と同じ意味で、最も簡略な記述を選べ。

$$i = i * 2;$$

1つ選択してください

i = \*2;

i = i + i;

i = 2 \* i;

i\*\*;

i \*= 2;

図 4: 確認テストの例

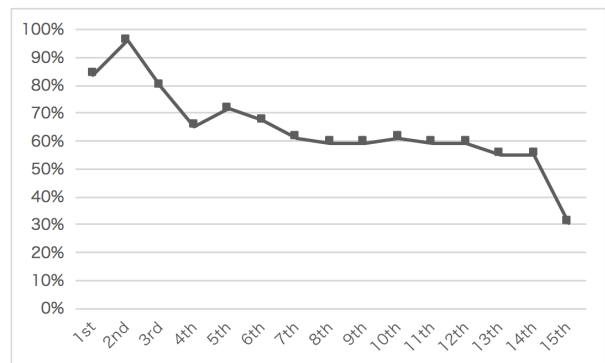


図 5: ビデオ視聴率の推移

割程度で推移している。15 回目が落ち込んでいるのは、期末テストのため通例の予習確認テストを行わず、試験範囲の復習に時間が充てられたこと、また大学の試験期間に重なったことが要因と考えられる。

各回ごとの視聴行動に関しては、授業日までの視聴回数を時間帯ごとに集計したものを図 6 に示す。この結果から、授業当日の授業開始直前、次いで当日の午前中に視聴数が多いことがわかる。また、アクセス元の IP アドレスから、視聴場所を自宅、大学、モバイルの 3 つに分類したものを図 7 に示す。時間帯を、授業当日の 12 時までと、それ以降の授業直前までとに分けているのは、授業直前は大学での視聴が主となることが予想されたため、それ以前の時間帯でどのような視聴方法が取られているか確認したかったからである。直前の駆け込み型の視聴を除けば、66%が自宅、15%がモバイルで視聴しており、時間外での予習ができていることがわかる。

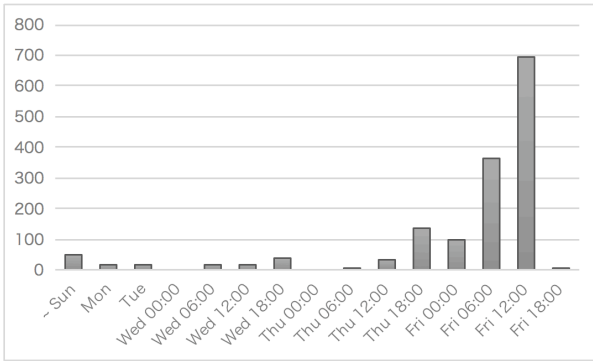


図 6: 予習ビデオの視聴時期

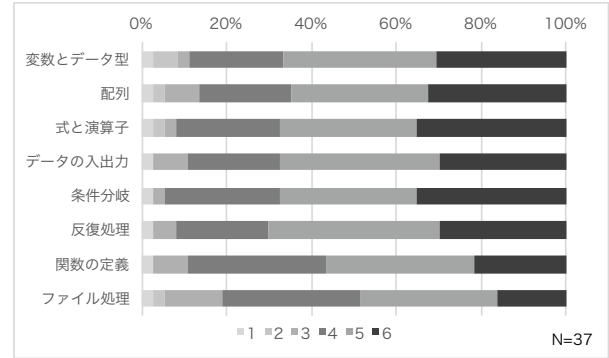


図 8: 項目ごとの達成度

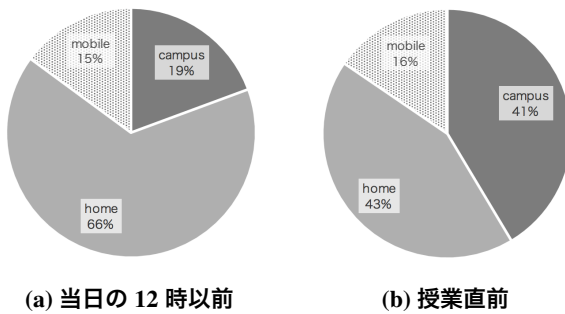
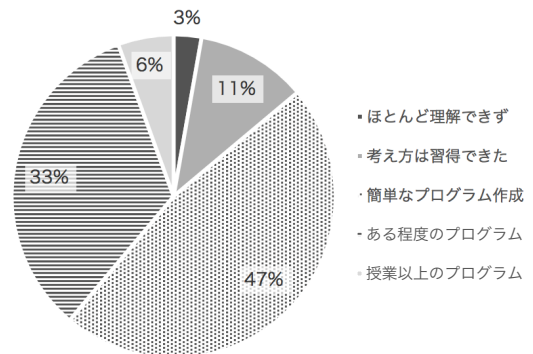


図 7: アクセス経路



N=36

図 9: C 言語の習得レベル

## 4.2 アンケート調査

毎回の授業後に学習項目ごとの理解度・達成度、また第 15 回の授業後に全般的な達成度や予習ビデオに関するアンケートを実施した。ここでは、第 15 回に実施したアンケートの結果を示す。図 8 は、C 言語の大きな学習項目ごとに達成度を自己評価してもらった結果である。評価は「1. 全く達成できていない」から「6. 完全に達成できた」までの 6 件法で行なった。結果として、すべての項目で 80%以上の受講者が達成できた（「ある程度」から「完全に」と回答しており、特にプログラミングの基本となる条件分岐や反復処理においては 90%以上となっている）。

また、授業を終えた時点での C 言語の習得レベルを、「ほとんど理解できず、身につかなかった」から「自分でも C 言語について積極的に勉強し、授業の例題以上のプログラムが作成できるようになった」の 5 段階から選んでもらったものを図 9 に示す。最も多かったのは、「C 言語の基本的な文法は習得し、簡単なプログラムが作成できるようになった」(47.2%)、次いで「授業の内容はほぼ理解でき、ある程度のプログラムが作成できるようになった」(33.3%)であった。

以上の結果から、基礎的なプログラミング科目としてはほぼ目標を達成していると考えている。

次に、今回反転授業の教材として導入した予習ビデオに関して、映像の長さは適当だったか (図 10)、実写による概要説明で、その回の目標が理解できたか (図 11)、スライド資料の部分にも、音声だけでなく教師の実写映像があった方がよいか (図 12) を尋ねた。

ビデオの長さについては、一般には 5~10 分程度が適当とされており、それ以上になると飽きてしまうと考えられる。本研究で作成したビデオは、ほとんどが 6~9 分であり、アンケートの結果でも「ちょうど良い長さ」との回答が 66.7%となっている。実際にビデオを編集した際には、説明の区切りのよいところで分割したため、項目によっては長くなってしまったものもあったが、概ね良好な評価であったと言える。

各回授業の概略説明のために作成した教師による実写映像については、目標が理解できたとの回答が合わせて 63.3%であったが、一方であまり理解できなかったという回答も 26.7%あり、よりわかりやすい説明とするため

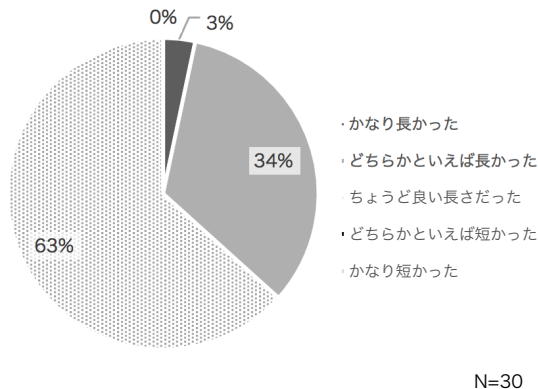


図 10: 予習ビデオの長さは適当だったか

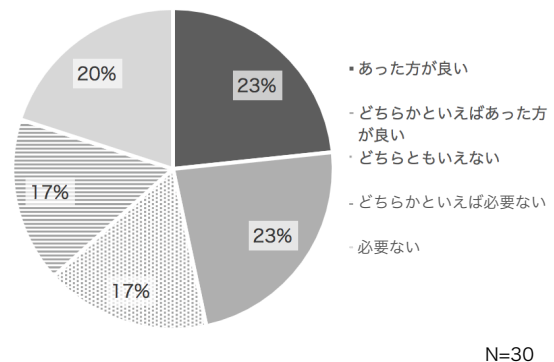


図 12: 教師の実写映像はあった方がよいか

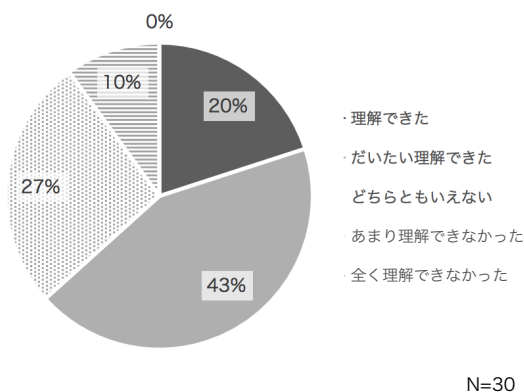


図 11: 各回の概要説明で目標が理解できたか

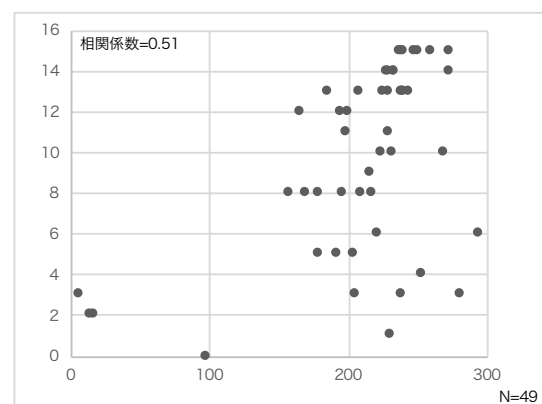


図 13: 視聴回数と成績との相関

に今後も改善したい。

今回の教材ビデオでは、本編は教師の映像はなく音声のみの説明とした。教師の実写映像による教材は、撮影や編集に時間と手間を要するだけでなく、映像の繋ぎ合わせて接続が不自然になるなど、品質の高い教材を作成するのは難しい。一般の大学の講義で継続的にビデオ教材を準備する場合、作成の手軽さと映像の品質はトレードオフの関係にあり、スライド資料に音声を収録する形式のものはコストパフォーマンスに優れていると考えられる。教材を視聴した受講者の意見では、教師の実写映像の有無は図 12 に示す通り大きく回答が分かれる結果となった。

その他、記述回答として「一番初めの先生の概要説明が楽しみなので、説明部分でも先生が直接出てほしい。」「現状だとスライド見るだけで何とかなるような予習ビデオの構成なので、見る価値のあるものにしてほしい。」などの改善提案があった。

### 4.3 成績との相関

最後に、予習ビデオの視聴行動と成績との相関を分析した。成績としては、プログラム作成課題、予習確認テスト、中間・期末テストをそれぞれ 100 点ずつとし、合計したものを用いた。また、各回のビデオを 1 度でも視聴したかどうかを累積し、すべて視聴した場合は 15 を視聴数とした。図 13 に、成績と視聴回数の相関を示す。相関係数は 0.51 で、やや相関が見られるという結果であったが、成績を高位・中位・低位の 3 階層に分けた上で集計したところ、図 14 に示すように特に高位群において視聴行動に幅が見られた。

そこでプログラミング経験ごとに視聴回数を集計したところ、図 15 に示すようにアプリケーション作成経験を持つ受講者についてはほとんど視聴していないことがわかった。すなわち、ある程度の経験がある受講者は、ビデオを視聴せず資料のみで予習したか、あるいはほぼ予習なしで経験のみで授業に臨んでいたことになる。授業としては、こういった経験のある学生に対しても視聴

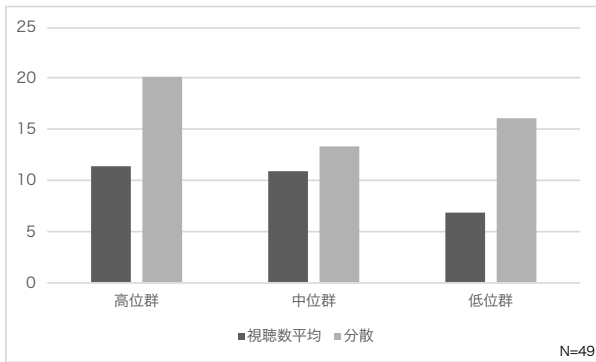


図 14: 成績群ごとの視聴数

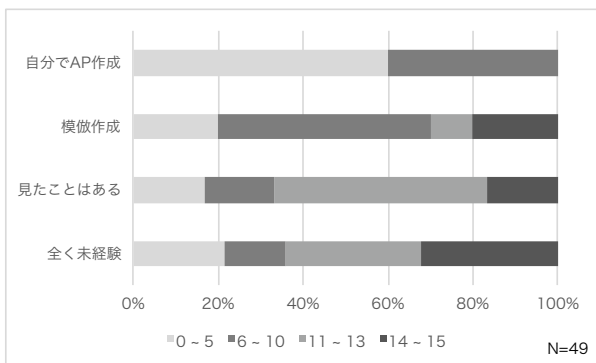


図 15: プログラミング経験と視聴数

の動機付けになるような工夫を今後検討する必要があると考えている。

## 5 まとめ

大学の情報系課程におけるプログラミング科目において、反転授業のための教材として予習ビデオを開発し、授業による実践と評価を行なった。その結果、基礎的なプログラミング能力を修得するという目標についてはある程度達成したと考えている。一方、反転授業の効果としては、教材と成績との相関分析、従来型の授業との比較などを含め、今後も検討して行く必要がある。

## 参考文献

- (1) ジョナサン・バーグマン, アーロン・サムズ, (山内祐平 訳): “反転授業”, オデッセイコミュニケーションズ (2014)
- (2) 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)
- (3) 喜多一, 岡本雅子: “写経型プログラミング学習と反転授業”, システム制御情報学会研究発表講演会講演論文集 60, 4p (2016)
- (4) 小島篤博, 真嶋由貴恵, 宮本貴朗, 青木茂樹: “e ラーニングを導入した全学情報教育における教育実践と質保証”, 教育システム情報学会研究報告 27(7), pp.213-218 (2013)

- (5) 林晴比古: “明快入門 C スーパービギナー編”, ソフトバンククリエイティブ, 東京 (2013)
- (6) 小島篤博, 青木茂樹, 宮本貴朗: “大学基盤システムと連携した Moodle による授業支援システムの構築”, 日本教育工学会 第 28 回全国大会, pp.229-230 (2012)
- (7) 青木茂樹, 小島篤博, 星野聡孝, 宮本貴朗: “出席管理システムの開発・運用と利用状況解析”, 電子情報通信学会論文誌 (D), J97-D(5), pp.1053-1057 (2014)
- (8) 小島篤博, 青木茂樹, 宮本貴朗: “大阪府立大学における Moodle のバージョンアップ”, 日本 Moodle 協会全国大会 (2015) 発表論文集, pp.44-47 (2015)
- (9) 林敏浩: “e-Knowledge コンソーシアム四国の教育クラウド運用と Learning Analytics への課題”, コンピュータ&エデュケーション, Vol.38, pp.49-54 (2015)