

プログラミング教育へのアクティブラーニング導入の検討

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
帝京大学理工学部情報電子工学科

概要

本稿では、帝京大学宇都宮キャンパスのアクティブラーニング推進ワーキンググループの情報電子工学科プログラミング教育サブワーキンググループ(SWG)の2015年度の活動の成果を報告する。本SWGは2015年7月から2016年3月までに、27回の会議を行い、その結果として、情報科学コースの教授内容とアクティブラーニング導入の方針を決定し、1年次の「プログラミング1」の具体的な授業設計を行った。また、本学科のプログラミング教育で共通に使用するルーブリックを作成した。作成したルーブリックや教材の試用と学生補助員養成を兼ねて在学生を対象にプログラミング勉強会を実施した。

1. はじめに

2015年度に帝京大学宇都宮キャンパスにアクティブラーニング推進ワーキンググループが発足し、そのサブワーキンググループ(SWG)として、情報電子工学科のプログラミング教育を検討するグループが設置された。メンバーは著者の7名である。一方、情報電子工学科では教育カリキュラム内にJABEE認定プログラムを設けることを目指している。JABEEとは日本技術者教育認定機構のことである。JABEEは第三者機関として大学等の高等教育機関で実施されている技術者を育成する教育プログラムを、国際的な同等性を持つ基準に基づいて認定している。

そこで、本SWGでは、JABEEの視点でプログラミング科目と実習科目の教授内容を検討するとともに、アクティブラーニング推進の視点でこれらの教授法について議論を行ってきた。本稿ではその成果を、以下を中心に報告する。

- ・ コースカリキュラムの設計
- ・ 「プログラミング1」の授業設計
- ・ プログラミング教育用ルーブリックの作成
- ・ プログラミング勉強会の実践

2. コースカリキュラムの設計

2.1 コースカリキュラムの全体

本学情報電子工学科では、JABEE認定を目指す教育プログラムを「情報科学コース」と呼び、他に情報メディアコース、エレクトロニクスコースの3コース制をとることとした。情報科学コースの教育カリキュラムは学科内のカリキュラム検討ワーキンググループで2014年11月から検討を行い、2015年7月には一通りの完成を見た。

本SWGでは具体的教授内容とアクティブラーニングを導入した教授法について検討を行い、図1に示すコースカリキュラムを設計した。文献[1]ではアクティブラーニングを、「一般的なアクティブラーニング」と「高次のアクティブラーニング」に大別している。図1はこの分類に従って科目を位置付けた。

高次のアクティブラーニングとはPBL(Project Based Learning)のような課題解決型の学習活動であり、文献[1]では「高次のアクティブラーニングを4年間連続的に配置することが重要である」としている。それは、「知識基盤社会の現代において重要とされる『知』とは専門知識の量ではなく、その専門知識を活用し課題解決する能力のことであり、この能力は繰り返すことによって獲得される能力である」からである。そこで、図1においても各学年に高次のアクティブラーニング科目を配置した。

Study on Introduction of Active Learning to Programming Education.

Hiro Yoshi Watanabe, Kozo Mizutani, Takuo Mori, Masayuki Arai, Shigeru Sasaki, Fumihito Furukawa and Kumiko Takai.

Department of Information and Electronics Engineering, Faculty of Science and Engineering, Teikyo University

授業形態	1年次		2年次		3年次		4年次	
	前期	後期	前期	後期	前期	後期	前期	後期
関連講義	論理数学	情報技術基礎	プログラミング言語論 データ構造とアルゴリズム コンピュータネットワーク	オートマトンと計算理論 計算機アーキテクチャ データベース論	ソフトウェア開発技法 情報システムデザイン オペレーティングシステム	ソフトウェア工学 プロジェクト管理 情報セキュリティ 人工知能		
AL主体授業	プログラミング1	プログラミング2	情報科学プログラミング1	情報科学プログラミング2	一般的アクティブラーニング			
実験実習		プログラミング演習1	基礎工学実験	情報科学基礎実習1 情報科学基礎実習2	ネットワーク演習 システム開発演習			
PBL		プロジェクト演習	プログラミング演習2		情報科学実習1	情報科学実習2		
ゼミ卒研	高次のアクティブラーニング					情報電子ゼミナール	卒業研究	

AL:アクティブラーニング, PBL:Project Based Learning

図1 情報科学コースカリキュラムへのアクティブラーニングの導入

一般的アクティブラーニングには、(1)講義を主体として能動的な学習活動を導入した授業、(2)反転授業のようにアクティブラーニング(AL)を主体として設計した授業、(3)知識定着のための実験実習があると考えられる。図1においては(2)アクティブラーニング主体授業と(3)実験実習を一般的アクティブラーニングと位置付けた。(1)の講義を主体として能動的な学習活動を導入する授業は、当面、各科目を担当する教員の工夫により進めることとし、図1ではアクティブラーニングとして明記していない。また、図1における関連講義はカリキュラムのごく一部である。サブワーキンググループでは、図1においてAL主体授業とPBLに位置付けた科目に関して教授内容と教授法を検討した。

2.2 PBL 科目

図1の1年次後期に配置した「プロジェクト演習」は、2008年度にヒューマン情報システム学科(現在の情報電子工学科)において設置したPBL科目である[2]。2年次前期に配置した「プログラミング演習2」では、チームまたは個人で独

自のプログラムを開発するようなPBLを行う授業を考えている。個人で開発する場合も、類似のプログラムを作成するメンバーで構成されるグループでの活動を取り入れる予定である。詳細は今後検討する。

3年次前期の「情報科学実習1」と3年次後期の「情報科学実習2」では、情報システム開発を対象とした本格的なPBLを行いたい。情報システム開発のPBLについては、既に優れた教育実践[3-5]があることから、今後これらを参考に授業設計を行う予定である。

2.3 プログラミング科目

これまでは1年次から2年次に「プログラミング1・2」、「応用プログラミング1・2」の4科目を配置しており、プログラミング言語はJavaを用いていた。これらの科目のうち「プログラミング1・2」は学科の必修科目として位置付けていた。

新カリキュラムでは、1年次の「プログラミング1・2」を学科必修科目として、2年次はコースごとに情報科学コースは「情報科学プログラミング1・2」を必修、情報メディアコースは「情報メディアプロ

プログラミング 1・2」を必修とした。その上で、学科必修の「プログラミング 1・2」ではプログラミング言語 Processing を用いることとし、「情報科学プログラミング 1・2」では Java を用いることとした。また、1 年次後期に「プログラミング演習 1」、2 年次前期に「プログラミング演習 2」を新設した。

これらのうち「プログラミング 1・2」と「プログラミング演習 1」については、具体的教授内容を決定した。また、「プログラミング 1」については教授法も検討を行った。

3. 「プログラミング 1」の授業設計

3.1 授業設計の方針

プログラミング系科目では講義に加えて、実際にプログラミングを行う演習の活動も入るため、元々、能動的な学習活動は含まれている。しかし、これまでの授業は、プログラミング力の修得という点で十分な成果があるとは言えない。また、中央教育審議会「高大接続答申」[6]などで示されている大学教育改革に沿って、「主体性を持って多様な人々と協働して学ぶ態度(主体性・多様性・協働性)を養うこと」に資する学習活動を取り入れる必要がある。

そこで、「プログラミング 1」の授業設計において次の方針をとることにした。

- (1) 第 1 の目的を、プログラミングに関する知識・技能を修得することにおく。
- (2) 主体性を持って多様な人々と協働して学ぶ態度形成に資するため、理解度の振り返り、他者とのコミュニケーションの機会の多い授業形態をとる。

自分で理解度を振り返る活動や他者とのコミュニケーションは(1)の知識・技能の修得にも大きく寄与すると考えられる。

こうした方針に基づき、具体的な方策として、反転授業の導入、ルーブリックの活用、学生補助員の活用を行うこととした。大学院生の補助員を TA(Teaching Assistant)、学部上級生の補助員を SA(Student Assistant)と呼び、今回は SA の活用を図る。

3.2 反転授業の導入

2015 年度の「プログラミング 1」において動画教材を用いた反転授業の導入を試みた[7]。反転授業や動画教材に対する学生の反応は肯定的であり、反転授業の導入に、一定の効果が認められた。

この知見を踏まえて、新カリキュラムでの「プログラミング 1」においても反転授業を導入することとした。現在のところ、事前学習として 45 分程度(15 分×3 本)のビデオ講義を視聴し、事前学習ワークシートに取り組むこととし、授業時間内は、次のような流れで行うことを考えている。

- (1) イントロダクションとステータス確認
- (2) 事前学習ワークシートに関するグループ活動
- (3) 授業内課題(個人活動)
- (4) 授業内課題(グループ活動)
- (5) 授業内課題の理解または発展課題(個人活動)
- (6) 振り返り課題
- (7) まとめ

これらの個人活動やグループ活動においては、後述する学生補助員(SA)が中心となって受講生をフォローする。グループ活動や SA とのやりとりが他者とのコミュニケーションの機会となる。

3.3 ルーブリックの活用

上述したような課題を中心とした授業においては、課題をこなしたり、課題を提出したりすることに注力してしまい、しっかり理解をすることが意識されなくなってしまうことがある。そこで、学習活動を振り返り、自分の理解を確認する活動を含めることとした。そのために、理解度を確認するための指標として、ルーブリックを作成した。これについては次の章で詳述する。

ルーブリックは次のように活用する予定である。

- ・ 毎回の授業の最後に振り返り課題に取り組み、その授業の学習項目に関するルーブリック(付録参照)を使って自分の理解度を把握するようにする。
- ・ 授業内課題や振り返り課題は、レベルの異なる複数の問題を用意しておき、対応するルーブリックのレベルを明示する。学生が自分のレベルにあった問題に取り組めるよ

うにするとともに、その問題をクリアできると、どのレベルをクリアできているかがわかるようにする。

- ・ 中間試験と期末試験において、ルーブリックの全体を使って自分の理解度を把握するようにする。

3.4 学生補助員(SA)の活用

授業においては、学生補助員(SA)に授業内課題のチェックや指導にあたってもらう。これまで、本学科のプログラミング授業では2名程度のSAを配置し、教室内を巡回して必要に応じて指導にあってもらおうような体制をとっていた。

札幌学院大学の森田らがSAを組織化し、授業運営のマニュアルを整備するなどして、SAが数名の学習者を担当するような形態での授業を行っている[8,9]。文献[8,9]には、そうしたSAによる指導体制を運用するポイントが紹介されている。これを参考にSAを活用した指導体制を構築する予定である。

4. プログラミング教育用ルーブリックの作成

ルーブリック(Rubric)は、縦方向に評価の観点、横方向に到達レベルあるいは尺度を持たせた表で、各セルに、各観点についてそのレベルの達成度を示す基準を具体的に記述したものである。縦方向を「規準」、横方向を「基準」と表すこともある。

作成したルーブリックを表1に示す。本学におけるプログラミング教育全体で使えるように、4年次になったときに達成できるレベルまでを段階化することとした。さらに上級もあり得るが、このルーブリックでは対象としない。

4.1 評価の観点(規準)の検討

ルーブリックの縦方向に配置する観点として、まず、「コーディング」と「コードリーディング」を設定した。これは、プログラムコードを書けるようになること、そのために他のプログラムコードを読んで理解できることが重要であるためである。

表1 プログラミング教育用のルーブリック

規準	レベル0	レベル1	レベル2	レベル3	レベル4
データ構造とアルゴリズム	【レベル1に満たない】	【基本的なデータ構造と処理を理解している】 ・データ構造にはどのようなものがあるか言える。 ・典型的な処理にはどのようなものがあるか言える。	【基本的なデータ構造と処理について深く理解している】 ・どのようなときに、どのデータ構造を使うか言える。 ・どのようなときに、どの処理を使うか言える。	【一般的な表現で処理手順を構成できる】 ・与えられた仕様に対するデータ構造を構成できる。 ・与えられた仕様に対するアルゴリズム・処理手順を書ける。	【問題を解決するデータ構造とアルゴリズムを分析できる】 ・与えられた仕様に対する複数のデータ構造、複数の処理手順をあげて、それぞれの長短を説明できる。
コーディング	【レベル1に満たない】	【基本的な文法を理解している】 ・1行程度の日本語での表現に対応するコードが書ける。 ・キーワード、記号など文法レベルのプログラムの空欄を埋めることができる。	【基本的な文法と処理手順の関連を理解している】 ・与えられたデータ構造と処理手順を基にプログラムコードを書くことができる。 ・条件や処理内容など、他との関連で考える必要があるプログラムの空欄を埋めることができる。	【基本的なプログラミングができる】 ・与えられた仕様に対して、データ構造と処理手順を自分で考えて、プログラムを作ることができる。	【実際のプログラミングができる】 ・問題状況に対して、プログラムの仕様を決めることができる。 ・与えられた仕様に対して、可読性が高く、正しく動作するプログラムを作ることができる。
コードリーディング	【レベル1に満たない】	【基本的な文法を理解している】 ・1行程度のコードを説明する日本語を書ける。	【基本的な文法と動作の関連を理解している】 ・プログラムの動作をトレースすることができる。	【プログラムコードを読んで理解できる】 ・コメント無しに、プログラムコードの仕様を推定することができる。	【実際のプログラムコードを理解し、説明できる】 ・大規模なプログラムのコードを読んで、プログラムの仕様や処理内容を、他者に説明できる。
テストとデバッグ	【レベル1に満たない】	【プログラムの実行とテストについて基本事項を理解している】 ・テストデータとは何か説明できる。 ・プログラムを入力し、(コンパイルし、)実行できる。 ・コンパイラ・インタプリタが報告する文法エラーを取り除くことができる。	【プログラムの実行とテストについて理解している】 ・与えられた仕様をテストするテストデータを構成できる。 ・容易に想定される原因による論理的エラーを取り除くことができる。 ・デバッグ機能を使ってプログラムの動作をトレースできる。	【基本的なテストとデバッグを行うことができる】 ・テストデータを使って、デバッグをすることができる。 ・デバッグ機能を使って、デバッグをすることができる。	【実際のテストとデバッグを行うことができる】 ・特殊な条件下で発生するなど、容易には想定できない原因による不具合を解消することができる。
可読性	【レベル1に満たない】	【プログラムの可読性について基本事項を理解している】 ・可読性が高いプログラムがどのようなものか言える。	【プログラムの可読性について理解している】 ・可読性を高めるため、なぜそうするのかを説明できる。	【可読性の高いプログラムコードを書くことができる】 ・読みにくいプログラムコードの可読性を高めることができる。 ・プログラミングにおいて可読性の高いプログラムコードを書くことができる。	【プログラムの可読性について分析できる】 ・可読性を高めるコーディングルールを決めることができる。

次に、正しく動作するプログラムを作成するための「テストとデバッグ」、読みやすいプログラムを作成できるようにするための「可読性」という項目も設けた。これらは、本来「コーディング」に含まれるが、コーディングで考慮すべき事項が多くなってしまうことから、別の観点として位置付ける方が望ましいと判断した。

さらに、プログラムを作成するために前提となる「データ構造とアルゴリズム」についての理解も観点に含めることにした。

4.2 到達レベル(基準)の検討

ルーブリックの横方向となる到達レベルの段階分けやその記述を検討する際に、教育目標の分類体系[10]や ICE モデル[11]が参考になる。これらとの関連を述べておく。

表 2 にブルームの分類体系とマルザーノらが構築した新しい分類体系[10]における 6 つのレベルを示す。レベル 1 が低いレベル、レベル 6 が高いレベルである。

作成したルーブリック(表 1)は 4 つのレベルとレベル 1 に満たないレベル 0 から成っており、レベル 1 からレベル 4 が概ね表 2 のレベル 1 からレベル 4 に対応する。具体的には、レベル 1 は文法等の知識を覚えるレベル、レベル 2 は例題等の理解ができていないレベル、レベル 3 は知識を応用して一通りのことができるレベル、レベル 4 は分析することでより高度に知識を活用できるレベルであり、作成したルーブリックは表 2 のブルームの分類体系に近い。

表2 教育目標の分類体系の 6 つのレベル

レベル	ブルーム	マルザーノ & ケンドール
6	評価	自律システム
5	総合	メタ認知システム
4	分析	知識の活用
3	応用	分析
2	理解	理解
1	知識	取り出し

次に ICE モデルとの関連を述べる。ICE モデルは、学びの段階を表したもので、I(Ideas)、C(Connections)、E(Extensions)の頭文字をとったものである。簡単に言うと、以下の三段階であるが、詳しくは文献[11]を参照されたい

- ・ I(アイディア):一つ一つの知識を学ぶ段階
- ・ C(つながり):学んだことと既に知っていることをつなげ深い理解に至る段階
- ・ E(応用):学びの最終段階で学んだことを十分に自分のものにしていく段階

作成したルーブリック(表 1)では、レベル 1 とレベル 2 が I(アイディア)の段階、レベル 3 が C(つながり)の段階、レベル 4 が E(応用)の段階に対応する。

以上のように、作成したルーブリックの到達レベルの段階分けは、教育目標の分類体系や広く用いられている ICE モデルに適合している。ただし、レベルを記述した文章表現については、今後の授業実践を通して改善していく必要がある。

4.3 学習項目ごとの評価

プログラミング全体としては表 1 のルーブリックを用いて評価を行うが、個々の学習項目についてもルーブリックを適用する。具体的には、各回の授業において学んだ学習項目について「コーディング」と「コードリーディング」の観点で、学生が自己評価を行う活動を取り入れる。

付録に、後述する「プログラミング勉強会」で使用したチェックシートの例を示す。ここに掲載されている学習項目は 1 年次のプログラミング教育において扱う予定の項目である。

5. プログラミング勉強会の実践

5.1 プログラミング勉強会の概要

これまでの検討を基にして、在学生を対象としたプログラミング勉強会を実施した。プログラミング勉強会の目標は次の通りである。

- ・ 授業の学生補助員(SA)の候補にプログラミング言語 Processing に慣れてもらうこと
- ・ 在学生にプログラミングの学び直しの機会を与えること

- 作成したルーブリック、問題、および教材の改善のための情報を得ること

最初に、1年次から3年次までの必修科目の授業の中でプログラミング勉強会を行うことを周知し、ガイダンスの日程を告知した。

ガイダンスは1/26(火)、1/27(水)の2回に分けて実施し、勉強会の進め方とプログラミング言語 Processing の概要を説明した後、参加を希望する者からルーブリックを使った自己評価シートを提出してもらった。他に、研究室に配属が決定した学生へガイダンスを行った研究室もあった。ガイダンスの参加者数を表3に示す。

表3 ガイダンスの参加者数

学年	1/26	1/27	研究室	合計
4年次	0	1		1
3年次	8(1)	1	9	18(1)
2年次	1	8(7)		9(7)
1年次	5	10(4)		15(4)
合計	14(1)	20(11)	9	43(12)

()内は学生補助員(SA)希望者数

プログラミング勉強会は、全体として以下の流れをとった。

- (1)ルーブリックによる事前の自己評価
- (2)LMSに掲載された教材・課題による学習
- (3)ルーブリックによる事後の自己評価・アンケートへの回答

事前の自己評価はガイダンス時に実施したが、未実施の学生には勉強会の最初に実施してもらった。勉強会は2/9(火)と2/10(水)の2日間で、都合が悪い学生には、どちらか一方の参加やeラーニング形式での学習も認めた。

LMSには次の学習項目について、教材と課題を掲載した。1つの学習項目はおよそ授業1回分の内容である。

- (1)データ型・演算子、(2)変数、(3)for文、(4)while文、(5)if文、(6)switch文、(7)二重ループ処理、(8)関数、(9)一次元配列、(10)二次元配列、(11)テキストファイルの入出力、(12)テスト駆動開発、(13)デバッガの使い方、(14)文

字列操作(課題のみ)

付録に示したルーブリックの学習項目、つまり、1年次で扱う学習項目はカバーすることができなかった。

勉強会の1日の時間割を表4に示す。おやつ時間は学生食堂の一角において、担当教員が用意したお菓子等をつまみながら、参加者の交流を図った。

表4 プログラミング勉強会の時間割

10:00 – 12:00	学習時間(1)
12:00 – 13:00	お昼休み
13:00 – 15:00	学習時間(2)
15:00 – 15:30	休憩(おやつ)
15:30 – 17:00	学習時間(3)

学習時間(1)から(3)では、個人またはグループで学習に取り組み、課題に対するプログラムを作成したら、教員や友人に確認してもらうように指示した。上で述べた学習項目は、どれをやってもよいことにし、事前の自己評価の結果を基に、苦手なところを選んで学習するとよいとアドバイスした。ただし、ほとんどの学生はLMSに掲載されている順に学習を進めていた。また、学習時間毎に学習した内容や疑問点・教材の誤字脱字・分かりにくい点などをLMSの日誌機能へ投稿するように指示した。

勉強会への参加数は、2/9は34名、2/10は20名であった。2日間とも出席した学生は18名であった。また、表3の人数のうちの9名はLMSのプログラミング勉強会へのアクセスが全く無かった。プログラミング勉強会の2日間の各学習時間には、3~5名の教員がコンピュータ教室に待機し、来年度に使用する問題や教材の作成をしながら、ときどき巡回を行う程度で、積極的な教授活動はしなかった。

5.2 アンケート結果

プログラミング勉強会後にアンケートを行った。アンケートへの回答者は17名であり、回答者は2日目の勉強会に参加した学生ばかりである。これは、LMS上のアンケートを回答可能な状態にした

のが、2日目の最後であったためだと思われる。回答者17名のうち2日目のみに参加した学生は2名であった。

勉強会に参加した動機(複数回答)の質問への回答を図2に示す。プログラミングの復習をしたいと言う学生が多いことがわかる。

プログラミング勉強会に関する質問への回答を図3に示す。有用性の質問、楽しさの質問とも全員がポジティブな回答であった。自由記述からは「課題の量もちょうど良く、学んだことを生かしてオリジナルのプログラムを組む時間ができたため楽しめた」「あたたかい感じで、自分のペースで進められたので勉強しやすかった」など自分で学べるのが良いという意見が4件、おやつや交流タイムが良かったとする意見が4件あった。これらのことから、プログラミング勉強会全体としては成功だったと言える。

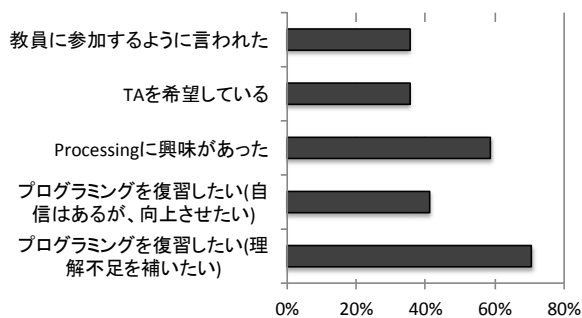


図2 勉強会に参加した動機

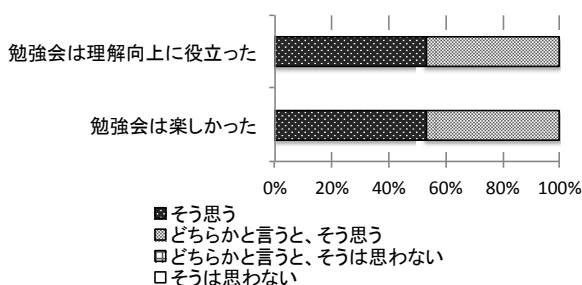


図3 プログラミング勉強会についての質問への回答

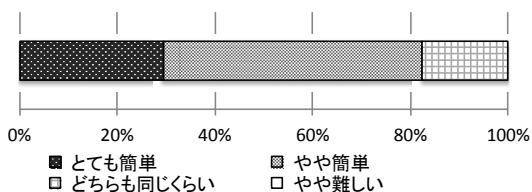


図4 ProcessingはJavaに比べて簡単かどうかについての質問への回答

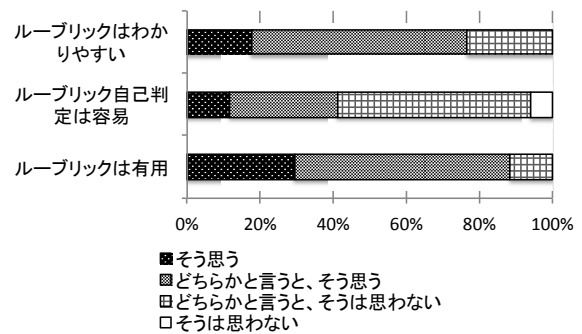


図5 ルーブリックについての質問への回答

Java言語に比較してProcessingはどうかという質問への回答を図4に示す。とても簡単、やや簡単と回答した学生が80%であったことから、Processingに変えることで1年次の学生が学習しやすくなることが期待される。一方で、2年次でJava言語へ移行する際に注意が必要である。

ルーブリックに関する質問への回答を図5に示す。ルーブリックはわかりやすいと感じる学生やルーブリックが有用であると感じる学生が多い。自由記述においても以下のような意見があった。

- 自分の能力確認するためにとてもいいと思った。また目標設定もルーブリックをもとに立てることが可能なのでよかった。
- 自分のプログラミングの苦手な部分を改めて確認することなどができて、ルーブリックは自分のプログラミングスキルチェックに有効だと思いました。

一方で、自分がルーブリックのどのレベルに該当するかを判断するのは難しいと感じる学生が多い。自由記述では以下のような意見があった。

- 評価は完全に自己判断なので、自分のレベルがどの程度と言えるのか迷うところがあった。
- 自信を持って、「3や4のレベルです」とは言いにくいと思う。

5.3 実践結果の考察

図3の結果から、プログラミング勉強会は全体として参加した学生にとって有益であり、満足度の高いものであったと言える。特に学生補助員(SA)希望ではなく、学び直しと意味で参加していた学生が多かったことから、このような勉強会のニーズ

があることも確認できた。一方で、学生の日誌への書き込みにおいて学習項目について自分が理解不足である旨の記述がいくつか見受けられた。今後同様な勉強会を実施する際には、学生の日誌への書き込みのタイミングで教員が書き込まれた内容に目を通して、フォローを行うようにすると、さらに効果が高まると考えられる。

来年度の授業の学生補助員(SA)候補についても12名の希望者を確保できたことから、当初の目的を果たすことができた。

図5の結果から、ルーブリックの有用性も確認できた。ルーブリック自体はわかりやすいが、自己判定が難しいという傾向があったことから、授業において教員による課題へのフィードバックやフォローを適切に行うことに留意する必要がある。今後、ルーブリックを用いた学生の自己評価の結果を分析し、全体的な理解度の把握と個々の学生へのフォローをどのように行うかを検討したい。

6. おわりに

本稿では2015年度の情報電子工学科プログラミング教育SWGの成果について報告した。2016年度は1年次の授業において反転授業とルーブリックを取り入れた授業実践を行うとともに、2年次以降の授業を具体的に設計する予定である。

参考文献

- [1] 河合塾編著, “「学び」の質を保証するアクティブラーニング -3年間の全国大学調査から-”, 東信堂, 2014
- [2] 荒井正之, 飽本一裕, 佐々木茂, 海上隆, 盛拓生, 近藤直樹, “ヒューマン情報システム学科におけるプロジェクト演習の授業設計と初年度実践報告”, ラーニングテクノロジー開発室年報, Vol.6, pp.79-86, 2009
- [3] 井上明, 金田重郎, “実システム開発を通じた社会連携型PBLの提案と評価”, 情報処理学会論文誌 Vol.49, No.2, pp.930-943, 2008
- [4] 糸野文洋, 辻村泰寛, 大木幹雄, 山地秀美, 石原次郎, 松田洋, “地域と連携した実践的ソフ

トウェア開発教育の試みとその効果”, 私立大学情報教育協会 平成24年度教育改革ICT戦略大会 資料, pp.208-209, 2012

[5] 大場みち子, 伊藤恵, “実システム開発PBLの実践事例”, 情報教育シンポジウムSSS2014論文集, pp.81-88, 2014

[6] 中央教育審議会:新しい時代にふさわしい高大接続の実現に向けた高等学校教育、大学教育、大学入学者選抜の一体的改革について(答申)(中教審第177号),

http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1354191.htm, 2016/03/31 アクセス

[7] 水谷晃三, 高井久美子, “プログラミング初学者を対象と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告, Vol.2015-CE-133, No.34, pp.1-8, 2015

[8] 森田彦, “学生教育補助員を活用した演習教育 —「プログラミング」の場合—”, 札幌学院大学総合研究所「社会情報」, Vol.14, No.2, pp.151-166, 2005

[9] 森田彦, “SAを活用した授業運営”, 札幌学院大学総合研究所「社会情報」, Vol.18, No.2, pp.117-129, 2009

[10] R.J.マルザーノ, J.S.ケンドール (黒上晴夫, 泰山裕 訳), “教育目標をデザインする 授業設計のための新しい分類体系”, 北大路書房, 2013

[11] S.F.ヤング, R.J.ウィルソン (土持ゲーリー法一 監訳), “「主体的学び」につなげる評価と学習方法 ~カナダで実践されるICEモデル~”, 東信堂, 2013

付録 学習項目毎のルーブリックの例

プログラミング勉強会事前スキルチェック(コーディング)

名前

規準	レベル0	レベル1	レベル2	レベル3	レベル4
コーディング	【レベル1のレベルに満たない】	【基本的な文法を理解している】 ・1行程度の日本語での表現に対応するコードが書ける。 ・キーワード、記号など文法レベルのプログラムの空欄を埋めることができる。	【基本的な文法と処理手順の関連を理解している】 ・与えられたデータ構造と処理手順を基にプログラムコードを書ける。 ・条件や処理内容など、他との関連で考える必要があるプログラムの空欄を埋めることができる。	【基本的なプログラミングができる】 ・与えられた仕様に対して、データ構造と処理手順を自分で考えて、プログラムを作ることができる。	【実的なプログラムを作ることができる】 ・問題状況に対して、プログラムの仕様を決めることができる。 ・与えられた仕様に対して、可読性が高く、正しく動作するプログラムを作ることができる。
データ型と演算子					
変数					
if文					
switch文					
for文					
while文					
二重ループ					
配列					
二次元配列					
関数					
クラス					
ファイル入出力					
文字列の処理					
オブジェクト指向					
LIST					
MAP					
描画					
マウス情報の使用					
タイマーの使用					
GUIライブラリ					