

SW융합인재 양성을 위한 비전공자 프로그래밍 학습에 관한 사례 연구

서주영
아주대학교 다산학부대학

A Case Study on Programming Learning of Non-SW Majors for SW Convergence Education

Jooyoung Seo
DASAN University College, AJOU University

요 약 최근 SW융합인재 양성을 위한 비전공자 대상의 SW교육에 관심이 높아지고 있다. 국내는 SW중심대학을 필두로 전공과 상관없이 모든 학생들에게 SW를 기초교육으로 필수화하는 추세이다. 본 논문은 프로그래밍 수업 사례를 통해 SW 전공자와 비전공자, 비전공자의 경우 인문 계열과 이공 계열의 차이가 프로그래밍 학습 능력과 느끼는 어려움에 연관을 보이는 지를 학업 성취도 분석과 설문 및 개인 상담을 통해 살펴보았다. 그 결과 전공에 따른 학업 성취도엔 유의미한 차이를 보이지 않았으나, 인문 계열이 이공 계열보다 실습, 과제, 팀프로젝트와 같은 실제 프로그램 구현에 대한 체감 난이도는 높게 분석됐다. 상담 결과로 프로그래밍 문제 자체에 대한 이해 부족, 학습과 과제에 도움을 줄 수 있는 친구나 튜터와의 관계 부족, 단편적 교과 지식만으로는 학습 동기를 높이기 힘들기 주요함을 알 수 있었다. 본 연구를 토대로 비전공자를 위한 SW교육 운영 방향에 대해 제언할 수 있을 것으로 기대한다.

주제어 : 비전공자 SW교육, SW융합, 프로그래밍, 학업 성취도, 어려움

Abstract Recently, there has been a growing interest in SW education for non-SW major in order to nurture SW convergence talent. In Korea, it is a tendency to make SW mandatory for basic education to all students regardless of their major, starting with SW-oriented universities. Through a case study of programming lesson, the paper compared differences in academic achievements and difficulties of learning between SW majors and non-majors and between the humanities department and the science department. As a result, although there was no significant difference in academic achievement according to majors, the humanities department had more difficulty in implementing programs such as practices, assignments, and team project. Through the interview, lack of understanding about programming problem itself, lack of relationship with friend or tutor that can help assignments, and difficulty in learning motivation by piecemeal curriculum knowledge alone were the main causes. The results will be expected to propose the direction of SW education for non-SW majors.

Key Words : Non-majors SW Education, SW Convergence, Programming, Academic Achievements, Difficulty

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음 (2015-0-00908)

Received 30 May 2017, Revised 30 June 2017

Accepted 20 July 2017, Published 28 July 2017

Corresponding Author: Jooyoung Seo(AJOU University)

Email: jyseo@ajou.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

최근 국내외적으로 초중고에서의 소프트웨어(SW) 교육에 대한 관심이 높아지고 있으며 국내의 경우 2018년도 중학교 교과과정을 시작으로 단계적으로 SW 교육을 의무화할 예정이다[1].

대학의 경우 기초교양과목으로의 SW 교육은 ‘프리젠테이션, 문서, 스프레드시트’ 작성의 오피스 도구 사용법을 다루는 ‘사무처리’ 교과목이나, ‘컴퓨터구조, 운영체제, 인터넷’의 컴퓨터 일반 개념을 다루는 ‘컴퓨터개론’ 교과목을 대표적인 SW 기초교육으로 생각해왔다. 이공 계열 학과의 경우 ‘프로그래밍’을 기초과목으로 선택 운영하기도 하지만 보편적 SW 교육으로 컴퓨터를 활용하는 교육을 더 일반적으로 여겨온 것이 사실이다[2,3,4].

최근엔 SW 융합 인재 양성이 강조되면서 이공 계열 학과뿐만 아니라 인문, 사회, 경영 등 다양한 전공 분야에서 SW 교육의 필요성이 대두되고 있으며, 특히 2015년부터는 SW중심대학을 필두로 전공과 상관없이 모든 학생들에게 기초교육으로써의 SW 교육을 필수화하려는 움직임이 크다[5,6].

SW 전공자와 비전공자의 SW 교육은 내용이나 난이도면에서 차이가 있을 수 있다고 생각해왔는데, 비전공자의 경우, 오피스 도구, 인터넷 사용과 같은 컴퓨터 활용 능력 학습만으로도 충분하다는 생각부터, 같은 교과를 배우더라도 전공자와는 학습 내용의 범위가 다르고, 더 쉬운 난이도로 운영되어야 한다고 생각하는 경우가 많다. 또는 스크래치(Scratch) 프로그래밍이나 컴퓨팅 사고력(Computational Thinking)과 같이 전공자를 대상으로 하는 SW 교육과는 다른 특별한 교육이 필요하다고 생각하는 경우도 일반적이다[2,3,7,8,9].

그러나 최근엔 창의적인 융합 인재 양성을 위해선 SW 비전공자들도 컴퓨터 과학에 대한 개념적 이해를 토대로 자신이 필요한 SW를 정의하고, 설계, 개발할 수 있는 능력도 갖추어야 한다는 인식이 커지고 있다 [5,7,10]. SW 융합 인재 양성과 더불어 보편적 SW 교육에 대한 필요성까지 대두되는 현시점에서 비전공자에 적합한 SW 교육의 운영 방향을 설정하기 위해선 SW 전공자와 비전공자, 비전공자의 경우 인문 계열과 이공 계열의 차이가 SW 학습에 어떤 차이를 보이는 지를 분석해 보는 것은 매우 중요하다.

본 논문에선 A대학에서 운영되고 있는 프로그래밍 교육의 수업 사례를 중심으로 SW 전공자와 비전공자의 프로그래밍 학습의 학업 성취도를 비교 분석한다. 현재 대학의 비전공자 SW 교육을 위해서 ‘비전공자들이 무엇을 학습해야하는 가’의 교과목 내용에 관해 많은 논의와 시도가 이뤄지고 있으나 비전공자의 프로그래밍 학습 능력 자체에 대한 분석은 미흡한 것이 사실이다. 본 사례 연구는 학습 내용보다 학생들의 학업 성취도를 살펴봄으로써 전공의 차이가 프로그래밍 수업을 소화해낼 수 있는 능력에도 차이를 보이는 지를 분석한다. 또한 프로그래밍 학습의 어려움에 대한 기존 연구들이 주로 ‘어떤 학습 내용을 이해하기 어려워하는지’에 관심을 갖는 것에 반해 본 사례 연구에선 비전공자들이 느끼는 체감 난이도를 분석하고 환경적 원인을 상담해봄으로써 심리적으로 비전공자들이 프로그래밍을 어렵게 여기는 배경에 대해서도 살펴본다.

본 사례 연구는 SW 교과목 중 대표적이라 할 수 있는 C 언어 프로그래밍 교과목을 대상으로 하며, 출석과 성적을 포함하는 수업 평가 자료를 토대로 학업 성취도를 분석하고, 수강생에 대한 설문과 개별 상담을 통해 비전공자들이 느끼는 프로그래밍 학습의 어려움을 분석한다.

본 논문의 구성은 다음과 같다. 2장에선 연구 배경과 함께 관련 연구를 살펴본다. 3장에선 본 논문의 핵심인 프로그래밍 수업에 관한 사례 연구를 살펴보고, 분석 결과로 SW 전공자와 비전공자의 차이, 비전공자의 경우 인문 계열과 이공 계열의 차이를 비교 분석한다. 4장에선 사례 분석 결과와 함께 이를 토대로 한 비전공자의 SW 교육의 운영 방향에 관해 제언한다.

2. 대학의 비전공자 SW 교육

2.1 비전공자 SW 기초교육

대학의 비전공자를 대상으로 하는 SW 교육은 보편적 SW 기초교육과 SW 융합교육으로 구분할 수 있다. 국내의 경우 SW중심사회를 선도할 인재양성을 위해 2015년부터 SW중심대학을 선정하여 지원하고 있으며, SW중심대학은 산업수요 기반의 전공교육 강화 및 비전공자 SW 교육, 소프트웨어 가치의 사회적 확산 등 종합적인 SW 중심 대학교육을 통해 SW 기반 문제 해결능력을 갖

준 다양한 인재 양성을 주요 사업 목적으로 한다[6]. 2015년에 8개 대학을 시작으로 2017년에 20개 대학, 2019년까지 30개 대학을 단계적으로 선정하여 지원할 예정이며, 타 전공지식과 SW 소양을 겸비한 융합인재를 2017년 22,000명, 2019년 97,000명 이상 양성할 것으로 기대하고 있다[11,12].

SW중심대학으로 선정된 대부분의 대학이 전공에 상관없이 모든 학생들에게 보편적 SW 기초교육을 목적으로 SW 관련 과목을 3학점 이상 필수로 이수하는 내용을 시행하고 있다[12]. <Table 1>은 2015년부터 2016년도까지 선정되어 비전공자 SW 기초교육을 운영 중에 있는 14개 SW중심대학의 비전공자 SW 기초교육 현황을 요약한 표이다[12,13,14,15,16]. <Table 1>의 SW중심대학 사례에서 보듯이 대학의 비전공자 SW 기초교육은 전교생이 공통 과목을 학습하거나, 계열 특성에 맞게 선정된 과목을 계열별, 전공별로 학습하는 양상을 보이고 있다.

전교생 대상으로 공통 과목을 운영하는 경우엔 ‘컴퓨팅사고(Computational Thinking)’ 역량과 관련된 과목을 선호하며 경북대, 고려대, 서강대, 성균관대, 충남대, 동국대, 서울여대, 한양대가 해당된다. 계열별 차이를 두고 운영하는 경우, 인문/사회 계열에 ‘데이터분석’ 역량, 자연/공과 계열의 경우 파이썬과 같은 대화형 프로그래밍 언어를 이용한 ‘프로그래밍’ 역량에 관한 과목을 선호하며, 가천대, 아주대, 부산대, 카이스트가 이러한 방식으로 운영 중에 있다. 국민대의 경우 전교생이 공통과목으로 프로그래밍 교과목을 선정하고 있는 점이 특이점이다.

컴퓨팅사고 역량은 ‘컴퓨터를 활용하여 해결 가능한 문제를 정의하고 해결책을 컴퓨터 시스템을 통해 효과적으로 수행하는 사고 과정’으로 ‘자료수집, 자료분석, 자료표현, 문제분해, 추상화, 알고리즘과 절차화, 자동화, 시뮬레이션, 병렬화’의 9가지 핵심 요소를 강조하고 있다 [17,18,19].

데이터분석 역량은 데이터로부터 새로운 가치를 발견하고, 과학적으로 검증된 데이터를 근거로 비즈니스 모델 결정, 정책 수립과 같이 다양한 분야에 활용 가능하기 때문에 최근 전공과 상관없이 관심이 매우 높다. 특히 SW와 관련이 없을 것으로 인식되던 인문, 사회 분야의 비전공자들에게도 갖추어야 할 유용한 SW 역량 중 하나로 각광 받고 있다.

프로그래밍 역량을 위해 C/C++, Java와 같은 전통적

인 프로그래밍 언어부터 최근엔 데이터를 다루는 파이썬과 R도 많이 학습하고 있다. SW 기반 문제 해결 능력을 학습하기 위한 교육용 프로그래밍 언어로 스크래치(Scratch), 엔트리(Entry)와 같은 이미지 및 블록형 프로그래밍 언어를 학습하기도 한다.

<Table 1> Status of SW basic education for non-majors in SW oriented universities

University	Grades	SW basic education
Gachon Univ.	3	(Select by majors) • SW-oriented world • Beginning of programming • Computational thinking • Hand-held programming • SW ecosystem • Application programming
Kyungpook Univ.	3 - 6	• (Common) SW problem solving • (Humanities) SW computational thinking • (Arts and physical) Introduction to the culture technology • (Science) Social network
Korea Univ.	3 - 6	• (Common) Computational thinking • (Depend on majors) SW coding education
Sogang Univ.	3 - 6	• (Common) Computational thinking • (Depend on majors) Advanced SW education
Sungkyunkwan Univ.	4	• (Common) Computational thinking and SW coding • (Depend on majors) Problem solving and algorithm
Sejong Univ.	3	(Select by majors) • Basics of SW • Introduction to the programming
Ajou Univ.	6	(Select by majors) • Computational thinking • Basics of data analysis • Programming
Chungnam Univ.	3 - 6	• (Common) Computational thinking (Select by majors) • Basics of programming • Mobile APP development • Web and Internet • Social network • Basics of data analysis • Basics of information security
Kookmin Univ.	3	• (Common) Computer programming (Excel, Scratch, Python)
Dongguk Univ.	6	• (Common) Computational thinking • Introduction to the programming

University	Grades	SW basic education
Pusan Univ.	3	<ul style="list-style-type: none"> • (Humanities) Bigdata (Arts) • Design SW • Physical computing (Science/Business) • Pin-tech • Python • Statistics SW and bigdata
Seoul Women's Univ.	3	<ul style="list-style-type: none"> • (Common) SW and creative thinking
KAIST Univ.	9	<ul style="list-style-type: none"> • Practical SW design and development process using open SW • Program language and package utilization education in IT Academy • Practice education using Elice SW training system
Hanyang Univ.	3	<ul style="list-style-type: none"> • (Common) Computational-X convergence talent training

2.2 비전공자 SW 융합교육

보편적 SW 기초교육과 달리 비전공자 SW 융합교육은 SW를 자신의 전공과 보다 적극적으로 결합하는 것을 목표로 하며, 대학의 경우 '부전공, 복수전공, 연계전공'으로 제공된다.

일반적으로 SW를 '부전공, 복수전공'으로 이수하려면 SW학과의 커리큘럼에 진입하기 전에 '수학, 과학'과 같은 기초 교과목을 필수로 이수해야만 한다. 이공 계열 학생의 경우, '수학, 과학'이 자신의 전공에서도 기초 교과목이기 때문에 바로 SW학과의 커리큘럼을 이수할 수 있지만, 인문 계열 학생의 경우 수학, 과학의 기초 교과목을 이수하는 것에만 1 ~ 2학기의 시간이 소요될 수밖에 없다. 이러한 이유로 인문 계열 학생의 SW학과 부전공, 복수전공 이수는 제도적으로도 매우 힘든 구조이다.

최근 국내에선 다양한 전공 배경을 갖는 SW 융합 인력을 양성하고자하는 S 기업의 지원 하에 '수학, 과학'과 같은 기초 교과목에 대한 장벽을 없애고, '이산수학'과 같은 SW학대에 필수적인 기초 교과목만으로 SW 전공 교과목에 진입할 수 있도록 하는 SW융합 인력 양성 과정이 25개의 대학에서 운영 중에 있다[20]. 이 과정을 운영하는 학교마다 커리큘럼엔 약간의 차이는 있지만 '이산수학, 컴퓨터개론, 프로그래밍언어, 컴퓨터구조, 자료구조, 알고리즘, 운영체제, 데이터베이스'의 교과를 공통으로 채택하고 학교에 따라 '소프트웨어공학, 융합세미나, 캡스톤' 등과 같은 SW학과 핵심 커리큘럼으로 30학점 이상 이수 시 부전공/복수전공의 학위를 수여한다.

이러한 움직임은 SW중심대학에도 이어져 부전공, 복수전공 뿐만 아니라 다양한 전공이 융합된 연계전공 구현에도 적극적이다. A대학의 경우, SW학과-기계공학과-산업공학과의 연계된 '자동차SW 연계전공'과 SW학과-사회과학대학-인문대학의 '인문사회데이터분석 연계전공'을 지원하고 있으며, B대학은 SW학과-전자공학과-기계공학과와의 '로봇SW 연계전공', SW학과-전기공학과와의 '에너지IoT 연계전공'을 운영 중이다[21,22]. '미디어아트SW 연계전공', '엔터테인먼트SW융합 전공'과 같이 예술계열학과와의 SW 융합교육도 시도되고 있다[23,24].

2.3 비전공자 프로그래밍 학습의 어려움

다양한 비전공자 SW 교육이 실시됨에 따라 비전공자 SW 교육의 어려움에 관한 연구도 관심이 높아지고 있다.

오미자(2017)는 스크래치를 이용한 프로그래밍 교육에 대한 비전공자의 인식 연구에서 실험에 참여한 비전공자 중 87.0%가 프로그래밍 학습에 어려움을 느꼈다고 답변하였고, 그 이유로 '프로그래밍 문법의 어려움(반복문, 함수), 코딩 시 발생하는 오류에 대한 수정 어려움, 프로그래밍에 대한 생소함, 관련 전공이 아니어서, 흥미 없음, 기타'의 설문 항목에 대해 '생소하고', '어려우며', '관련 전공도 아니다'라는 복합 답변을 한 비율이 1위를 차지하였고, 2위로 생소함, 3위로 문법의 어려움이 차지하였다고 보고하고 있다[25].

김수환(2015)의 컴퓨팅사고 교육에서 나타난 비전공 학습자의 어려움 분석 연구에선 '변수와 리스트의 학습, 아이디어를 생각하고 구현하는 과정, 적합한 명령어의 사용' 순서로 학습의 어려움을 언급하고 있다[2].

김경미외(2014)의 연구에선 파이썬 프로그래밍을 통한 코딩 수업의 난이도에 대해 여학생과 고등학교 때 문과 계열인 학생이 프로그래밍 수업에 대해 어려움을 느끼는 것으로 분석하였고, 그 원인으론 '문제 해결을 위한 논리적 사고의 어려움과 프로그래밍 개념 이해'로 분석하고 있다[5].

프로그래밍 학습의 어려움에 관한 기존 연구들이 학습 내용과 관련하여 그 원인을 찾으려는 분석이 많았던 반면, 학생들의 전공 차이와 연관된 학습 환경적 측면에서의 어려움에 대한 조사는 미흡하다. 본 사례 연구에서는 전공의 차이가 전통적인 컴퓨터 프로그래밍 수업을 소화할 수 있는 능력에도 차이를 보이는 지를 분석함

과 동시에 설문과 개별 상담을 통해 비전공자들이 심리적으로 프로그래밍을 어렵게 생각하는 배경에 대해서도 살펴보는 것을 목적으로 한다.

3. 사례 연구

3.1 연구 문제

대학의 전공 차이가 프로그래밍 수업을 이해하는데 영향을 주는 지를 다음 관점에서 분석한다.

- SW 전공자와 비전공자 뿐 만 아니라 비전공자의 경우 인문 계열과 이공 계열로 나누어 전공에 따른 학업 성취도의 차이를 살펴본다.
- 인문 계열과 이공 계열의 전공 차이가 프로그래밍 학습에 느끼는 어려움에 차이를 주는 지를 분석하고 그 원인을 살펴본다.

3.2 연구 대상

연구 대상은 <Table 2>의 교과과정이 동일하게 운영된 프로그래밍 수업을 수강한 SW 전공자 45명과 비전공자 21명이다. 비전공자는 인문 계열(상경대학, 사회과학대학, 인문대학) 12명, 이공 계열(자연대학, 공과대학) 9명의 분포로 구성되었다. 학업 성취도는 기본적으로 학생 개인이 갖는 교과목에 대한 관심과 학습 의지에 영향을 받을 수 있기에 본 연구에선 비전공자의 경우도 SW를 부전공으로 선택한 학생들을 대상으로 선정함으로써 SW에 대한 관심과 프로그래밍에 대한 학습 의지는 전공자와 유사하다고 가정한다.

다음 <Table 2>는 본 연구를 위해 선정된 프로그래밍 교과목의 수업계획안이다. 과목은 16주로 동일한 교수와 학습 교재 및 내용으로 운영되었으며, 출석, 개인 과제 2회, 팀 프로젝트 1회, 중간 및 기말고사의 시험 2회로 평가되었다.

<Table 2> Programming Course Syllabus

Items	Description
-------	-------------

Curriculum	<ul style="list-style-type: none"> • SW development process • Standard I/O, File I/O • Variable • Data type • Expression • Operators • Condition statement • Repetition statement • Function • Array • String • Pointer • Structure, union • Pointer application
Practice	14 times
Assignment	2 times
Team project	1 time
Exam	2 times (1 time practical exam)

3.3 연구 설계

SW 전공자와 비전공자의 프로그래밍에 대한 학업 성취도 차이를 비교 분석하기 위해 출석과 다양한 성적 항목을 포함한 수업 평가 결과 자료를 토대로 학업 성취도와 수업 참여도를 분석한다. 또한 수업에 대해 체감하는 난이도를 설문 조사를 통해 분석한다.

3.3.1 학업 성취도

수업 운영 자료 중 성적 평가에 관련하여 ‘실습, 과제, 팀 프로젝트, 중간 및 기말고사의 시험’의 각 항목과 출석 점수까지 포함한 ‘종합 성적’으로 학업 성취도를 분석한다.

3.3.2 수업 참여도

수업 참여도는 수업 운영 자료 중 ‘출석, 과제 제출’과 같이 수업에 참여한 정도로 측정한다.

3.3.3 체감 난이도

비전공자들이 실제 느끼는 프로그래밍 학습의 어려움을 살펴보기 위해 이론과 실습으로 구분하여 5점 척도로 체감하는 수업의 난이도를 점수화하여 조사한다. 1점부터 체감하는 난이도가 ‘매우 어려움’, 2점은 ‘어려움’, 3점은 ‘보통’, 4점은 ‘쉬움’, 5점은 ‘매우 쉬움’을 의미한다.

3.4 결과 분석

SW 전공자와 비전공자, 비전공자의 경우 인문 계열과 이공 계열로 분류하여 학업 성취도의 차이를 독립표본

t-검정으로 분석하였다. <Table 3>의 분석 결과를 보면 SW 전공자, 비전공자의 학업 성취도 차이는 유의수준 .05에서 '실습'을 제외하곤 유의미한 차이는 없으며, 특히 '종합 성적' 면에선 p값이 0.114로 유의수준 .05에선 SW 전공자, 비전공자의 학업 성취도는 유의미한 차이가 없는 것으로 나타났다.

<Table 3> Academic achievement of SW-majors/non-majors

Comparison items	Average		Standard Deviation		t value	p value
	M	N	M	N		
Practice	85.46	94.14	17.85	6.50	2.154	0.035
Assignment	64.89	69.13	26.00	20.76	0.654	0.515
Team project	64.96	73.56	16.72	18.69	1.875	0.065
Exam	45.21	55.72	21.73	23.68	1.777	0.080
Final grade	64.84	71.51	16.17	14.73	1.602	0.114

(M: SW majors, N: SW non-majors)

실습 항목에서 유의미한 차이를 보이는 것은 실습은 수업 시간에 이루어지는 학습 형태로 수업 참여도와 밀접한 관계를 갖기 때문으로 해석된다. 다음 <Table 4>의 수업 참여도 결과를 살펴보면 비전공자의 수업 참여도는 평균 99.48% 특히 출석율은 98.96%이며 SW 전공자의 전체 참여율 94.75%와 출석율 97.71%에 비해 평균적으로 높게 나타났다.

<Table 4> Class participation of SW-majors/non-majors

Comparison items	M	N
Practice	97.71%	98.96%
Assignment	91.79%	100%
Class participation	94.75%	99.48%

<Table 5>의 비전공자의 계열별 학업 성취도 차이의 분석 결과를 보면, 이공 계열이 인문 계열보다 실습을 제외한 모든 측면에서 평균이 높게 나왔으나 통계적으로 유의수준 .05에서 팀 프로젝트를 제외한 나머지 항목에선 유의미한 차이가 없었다. 종합 성적에선 p값이 0.103으로 유의수준 .05보다 높음으로 계열별 학업 성취도 면에서도 유의미한 차이가 없었다. 팀 프로젝트 항목에 대해 인문 계열과 이공 계열이 유의미한 차이를 보이는 것에 관해선 3.5.3 상담 의견을 통해 살펴보겠다.

<Table 5> Academic achievement of Humanities/Science-and-Engineering

Comparison items	Average		Standard Deviation		t value	p value
	H	S	H	S		
Practice	94.33	93.90	5.28	8.20	0.1445	0.887
Assignment	62.87	77.47	22.81	15.00	-1.665	0.112
Team project	65.84	83.85	16.21	17.42	-2.442	0.025
Exam	49.50	64.02	25.16	19.92	-1.426	0.170
Final grade	66.95	77.57	15.09	12.52	-1.712	0.103

(H: Humanities departments, S: Science-and-Engineering departments)

설문을 통한 비전공자의 프로그래밍 수업에 대한 체감 난이도 분석 결과는 <Table 6>과 같다. <Table 6>은 유의수준 .05에서 이론 수업에선 p값이 0.849로 체감 난이도에 유의미한 차이가 없음을 보여주고 있으나 실습 수업에선 p값이 0.023으로 인문 계열과 이공 계열 간에 실습 수업에 대한 체감 난이도엔 유의미한 차이가 있음을 알 수 있다.

<Table 6> Programming difficulties of Humanities/Science-and-Engineering

Comparison items	Average		Standard Deviation		t value	p value
	H	S	H	S		
Theory class	3.67	3.75	0.89	1.04	-0.193	0.849
Practice class	3.58	4.29	0.79	0.49	-2.482	0.023

결론적으로, 동일한 교수자와 교재를 기반으로 동일한 실습, 과제와 시험과 같이 동일한 내용과 난이도로 학습하고 평가했음에도, 프로그래밍 교과목에 대한 학습 성취도 측면에선 SW 전공자, 비전공자는 유의미한 차이를 보이지 않았으며, 인문 계열과 이공 계열도 유의미한 차이를 보이지 않았다. 그러나 학습에 대한 체감 난이도는 이론 수업에 대해선 유의미한 차이를 보이지 않으나, 실습 수업에선 인문 계열과 이공 계열 간엔 유의미한 차이가 있음을 알 수 있었다. 특히 흥미로운 것은 <Table 5>와 같이 실습에 대한 평가 점수는 평균적으로 인문 계열이 이공 계열보다 높았음에도 <Table 6>에서 실습에 대해 느끼는 체감 난이도에선 인문 계열이 이공 계열보다 어렵다고 느꼈다는 점이다.

3.5 설문 조사 및 개인 상담

비전공자가 실제 체감하는 프로그래밍 학습의 어려움과 계열별 차이에 대한 분석을 위해 비전공자를 대상으로 <Table 7>과 같은 설문 조사와 함께 개별 상담을 실시하였다.

<Table 7> Questionaries

No.	Questionaries
1	What type of problem should be more important in the base, advanced, and applied problems for practice?
2	How much time do you spend on self-study on average per week
3	How much time do you spend on average per assignment
4	What helped you most in solving assignments
5	Do you need the support of teaching secondary resources such as e-learning, instructional videos, tutor for self-study

3.5.1 설문 대상

설문 대상은 앞서 <Table 2>의 동일 교과과정으로 1년간 운영된 2개 반의 비전공자 수강생 총 48명을 대상으로 인문 계열 27명, 이공 계열 21명의 분포로 구성되었다.

3.5.2 설문 결과

설문 문항 1의 답변을 분석하면 <Table 8>과 같이 실습에서 인문 계열은 기초 문제가 강화될 필요가 있다고 72.73%나 답변하였고, 응용 문제의 경우 9.09%에 지나지 않았다. 반면, 이공 계열의 50%는 응용 문제가 더 필요하다고 답변하였다. <Table 6>의 실습의 체감 난이도에 대한 분석 결과와 함께 보면 어렵다는 답변이 높았던 인문 계열의 경우 기초를 다질 수 있는 문제를 좀 더 필요로 했고, 쉽다는 답변이 우세했던 이공 계열의 경우 기초 보다는 응용 문제로 실습의 난이도를 좀 더 높여야 된다는 응답을 한 것을 알 수 있다.

<Table 8> What type of problem should be more important in the base, advanced, and applied problems for practice?

	Basic	Advanced	Applied

문항 2의 자율 학습에 할애하는 시간을 살펴보면, <Table 9>와 같이 인문 계열과 이공 계열이 모두 주당 3-10시간이 44.45%, 53.39%로 할애하는 학습 시간의 분포가 유사했다. 반면 문항 3의 과제에 소요되는 시간을 살펴보면 <Table 10>과 같이 인문 계열과 이공 계열이 3일 이내에 과제를 해결하는 경우가 각각 66.66%와 71.43%로 유사하였으나 인문 계열의 경우 1주일 이상 소요된다고 답변한 경우도 22.23%로 높아서 개별 차이가 많이 남을 알 수 있었다. 이공 계열의 경우 1주일 이상 소요되는 경우는 없었다. 문항 2와 3의 설문 결과를 종합해 보면 계열별로 학업 성취도와 학습에 할애하는 시간에 차이를 보이지 않았으나, 팀프로젝트와 실습 수업에 대한 체감 난이도가 높았던 인문 계열은 과제에 할애하는 시간도 이공 계열에 비해 더 많은 시간을 투자하고 있음을 알 수 있었다.

<Table 9> How much time do you spend on self-study on average per week

	< 3hours	3-10hours	> 10hours
H	44.44%	44.45%	11.11%
S	33.33%	53.39%	14.29%

<Table 10> How much time do you spend on average per assignment

	< 3day	4-7days	> 1week
H	66.66%	11.11%	22.23%
S	71.43%	28.57%	0%

설문 문항 4의 과제 수행 시에 어떤 도움을 받았는지의 의견을 살펴보면 <Table 11>과 같이 인문 계열의 경우 과제 문서 > 인터넷 검색 = 교수의 순으로 도움을 받은 경우가 많았다. 이공 계열의 경우 친구 > 인터넷 검색 > 과제 문서 순으로 나타났으며, 인문 계열에선 낮게 나왔던 ‘친구의 도움’이 가장 많은 답변으로 조사된 것이 특이점이다.

<Table 11> What helped you most in solving assignments

또한 설문 문항 5의 튜터와 같은 수업 보조 자원에 대한 필요성에 대한 답변에서도 <Table 12>와 같이 인문 계열의 81.48%가 필요하다는 답변을 한 반면 이공 계열은 65%가 필요하지 않다는 상반된 답변을 하였다.

<Table 12> Do you need the support of teaching secondary resources such as e-learning, instructional videos, tutor for self-study

	Yes	No
H	81.48%	18.52%
S	35.00%	65.00%

3.5.3 상담 의견

설문 결과 중 다음의 특이점을 중심으로 비전공자들이 느끼는 프로그래밍 학습의 어려움에 대한 개별 상담을 실시하였다.

- 학습에 할애하는 시간의 설문 결과는 인문과 이공 계열의 시간 분포가 유사한 반면, 과제에 관해선 인문 계열은 개인별 차이가 크다.
- 과제를 해결하는 과정에서 이공 계열은 친구의 도움을 받는 비중이 높은 것과 달리 인문 계열은 친구의 도움을 받는 비중이 매우 낮다.
- 수업 보조 자원의 경우 인문 계열은 적극 필요하다고 답변한 반면 이공 계열은 필요하지 않다는 답변이 우세하다.

다음은 상담 의견을 종합한 내용이다.

첫째, 인문 계열의 대다수가 실습이나 과제에서 다루지는 문제 자체가 이해하기 어렵다는 의견을 피력했다. 실습 문제가 프로그래밍 이론 시간에 배우는 문법이나 기법을 학습하는 용도로는 괜찮으나 다루지는 기법이 왜 필요한지, 어떤 문제를 해결할 수 있는 지 등 문제 본연에 대한 설명은 부족하고 프로그래밍 문법 설명에만 집중되는 경향이 있으며, 이러한 학습 방식은 교과목을 단편적 지식으로만 받아들여지게 하고 학습 동기 부여도 저해한다고 응답하였다.

둘째, 사례 연구에서 이공 계열이 인문 계열보다 팀 프로젝트에서 우월한 평가를 받은 이유도 프로그래밍 수업에서 다루는 문제 경향이 이공 계열의 문제 경향과 유사한 측면이 있기에 팀 프로젝트 주제를 선정하고 프로그

램으로 구체화하는 과정에서 어려움이 크지 않았다고 응답하였다. 반면 인문 계열의 경우 주제 선정과 이를 프로그래밍 가능한 문제로 정의하는 단계가 너무 어려웠으며 이를 위해 프로젝트 초기에 시간을 많이 소모해서 실제 구현의 완성도가 떨어질 수밖에 없었다고 답변하였다.

셋째, 이공 계열의 경우 동아리 선후배를 포함하여 주변에 SW 전공자나 프로그래밍을 배운 친구들이 많기에 학습 도중 질문이 생기는 경우 쉽게 도움을 받을 수 있었다고 응답하였다. 실제 사례 연구의 학업 성취도 결과를 보면 실습 점수는 유사하나, 과제/프로젝트/시험 점수에선 이공 계열이 평균적으로 우수하게 나타났다. 종합해 보면 이공 계열은 현 수업 난이도에 만족하며, 강의실 내에서 수업 내용을 충분히 숙지하고 주변에 SW 전공자나 프로그래밍 유경험자와 쉽게 교류할 수 있기에 별도의 수업 보조 자원에 대한 요구도 크지 않았다. 반면 대다수의 인문 계열은 주변 도움이 어렵기에 과제 진척에 도움을 줄 튜터나 수업 보조 자료가 절실히 필요하다고 응답하였다.

4. 결론

본 논문에서 컴퓨터 프로그래밍 수업 사례를 통해 SW 전공자와 비전공자, 비전공자의 경우 인문 계열과 이공 계열의 차이가 프로그래밍 학습 능력과 어려움을 느끼는 정도에 연관을 보이는 지를 학업 성취도 분석과 설문 및 개인 상담을 통해 살펴보았다. 연구 결과는 독립표본 t-검정 분석을 통해 이루어졌으며 SW 전공자와 비전공자의 학업 성취도 분석 결과는 유의수준 .05에서 p값이 0.114로 유의미한 차이가 없었다. 비전공자의 경우, 인문 계열과 이공 계열의 학업 성취도 역시 0.103으로 유의미한 차이가 없었다.

그러나 비전공자를 대상으로 실시한 프로그래밍 학습의 체감 난이도에 대한 분석 결과는 이론 수업에선 0.849로 체감 난이도에 유의미한 차이가 없음을 보여주고 있으나 실습 수업에선 0.023으로 인문 계열과 이공 계열 간에 실습 수업에 대한 체감 난이도엔 유의미한 차이가 있으며 인문 계열이 프로그래밍 실습을 어렵게 생각함을 알 수 있었다. 실제 설문을 통해서도 수업을 이해하기 위한 자율 학습에 할애하는 시간은 계열에 상관없이 유사

하였으나, 과제를 해결하는 데에는 인문 계열이 이공 계열보다 더 많은 시간을 할애하고 있었고, 팀프로젝트에 대한 학업 성취도 역시 인문 계열이 이공 계열보다 낮게 분석되었다. 인문 계열이 이공 계열보다 프로그래밍 실습, 과제, 팀프로젝트와 같은 실제 프로그램을 구현하는 측면에 대한 체감 난이도가 높은 원인을 알기 위해 실시한 개별 상담 결과로 수업에서 다루지는 프로그래밍 문제 자체에 대한 이해 부족, 학습과 과제에 도움을 줄 수 있는 친구나 튜터와의 관계 부족, 단편적 교과 지식만으로는 학습 흥미와 동기를 높이기 힘들어 주요함을 알 수 있었다.

본 사례 연구 결과는 학습 능력엔 유의미한 차이를 보이지 않음에도 높은 체감 난이도를 보이는 비전공자들을 대상으로 하는 SW 교육의 수업 운영에 있어 다음과 같은 점들이 고려되어야 함을 시사하고 있다.

첫째, 학습 내용, 특히 실습 문제에 대해 구체적 현실 사례나 수강 대상의 전공에 맞춘 문제를 제시함으로써 문제를 이해하는 시간을 줄이고, 학습 흥미도 부여할 수 있는 수업 운영이 필요하다.

둘째, 학습한 내용과 관련하여 자신의 전공과의 연관성이나 융합의 가능성에 대해 스스로 고찰할 수 있도록 수업 이후 회고 시간을 운영하거나, 팀 프로젝트를 통해 관심 주제에 응용할 수 있는 수업 운영이 필요하다.

셋째, 수업 외 시간에 학습한 내용에 대한 질의응답을 도와줄 튜터와 비전공자들도 참여 가능한 SW 동아리를 양성함으로써 비전공자들도 SW 경험이 많은 사람들과 인적 관계를 쉽게 맺을 수 있도록 하는 제도적 지원이 필요하다.

위의 시사점에 대한 향후 연구로 비전공자 대상 교육에 있어서 수강 대상의 전공에 맞춘 문제 중심의 SW 교육의 필요성과 그 효과에 대한 연구를 진행 중에 있다. 특히 최근 다양한 분야에서 활용도가 높은 ‘데이터 분석’ 학문을 교육함으로써 인문 계열 학생들이 자신의 전공 문제를 ‘데이터 분석’의 SW기술로 해결해나가는 학습 과정을 통해 SW 교육의 체감 난이도를 낮추고 학습 흥미와 SW 융합의 동기를 높일 수 있는 지를 분석하는 연구를 현재 진행 중이다.

ACKNOWLEDGMENTS

This research was supported by the MISP(Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & communications Technology Promotion)(2015-0-00908).

REFERENCES

- [1] Ministry of Education, Ministry of Science, ICT and Future Planning, “Plan for Human Resource Development for SW-oriented Society”, 2015.
- [2] S. H. Kim, “Analysis of Non-Computer Majors’ Difficulties in Computational Thinking Education”, The Journal of Korean Association of Computer Education, Vol. 18, No. 3, pp.15-23, 2015.
- [3] S. H. Park, “Study of SW Education in University to enhance Computational Thinking”, Journal of Digital Convergence, Vol. 14, No. 4, pp.1-10, 2016.
- [4] S. Y. Pi, “A Study on Coding Education of Non-Computer Majors for IT Convergence Education”, Journal of Digital Convergence, Vol. 14, No. 10, pp.1-8, 2016.
- [5] K. M. Kim, H. S. Kim, “A Case Study on Necessity of Computer Programming for Interdisciplinary Education”, Journal of Digital Convergence, Vol. 12, No. 11, pp.339-348, 2014.
- [6] Ministry of Science, ICT and Future Planning, “Plan for SW-oriented University”, 2015.
- [7] J. E. Na, “Computational Thinking Education, Curriculum Development, Its Status”, Computational Thinking Forum, pp.45-51, 2015.
- [8] G. S. Ryu, “Development of Educational Model for ICT-based Convergence Expert”, Journal of the Korea Convergence Society, Vol. 6, No. 6, pp.75-80, 2015.
- [9] J. H. Ku, “Designing an App Inventor Curriculum for Computational Thinking based Non-majors Software Education”, Journal of Convergence for Information Technology, Vol. 7, No. 1, pp.61-66, 2017.

- [10] Ministry of Science, ICT and Future Planing, "Current Status and Implications of SW Education in the Major Countries", 2015.
- [11] SW oriented University Home page, SW oriented University Operation Plan, <https://www.software.kr/um/um05/um0501/um050102/um05010203.do>, May 29, 2017.
- [12] SW oriented University Home page, Overview of the Participating Universities of SW oriented University, <https://www.software.kr/um/um02/um0207/um020701.do>, May 29, 2017.
- [13] SW oriented University Home page of Gachon University, <http://xsw.gachon.ac.kr/cmd>, May 29, 2017.
- [14] SW oriented University Home page of Korea University, <http://helloworld.korea.ac.kr>, May 29, 2017.
- [15] Sungkyun SW Education Institute Home page, <http://ssen.skku.edu>, May 29, 2017.
- [16] SW oriented University Home page of Chungnam University, <http://wise.cnu.ac.kr>, May 29, 2017.
- [17] K. S. Oh, S. J. Ahn, "A Study on Development of Educational Contents about Computational Thinking", The Journal of Korean Association of Computer Education, Vol. 19, No. 2, pp.11-20, 2016.
- [18] H. Y. Jung, "An Empirical Study on Information Liberal Education in University based on IT Fluency and Computational Thinking Concept", Journal of The Korea Society of Computer and Information, Vol. 19, No. 2, pp.263-274, 2014.
- [19] J. M. Wing, "Computational Thinking", Communication of ACM, Vol. 49, No. 3, pp.33-35, 2006.
- [20] The Korea Economic Daily, "Samsung, Link to 25 Universities, Human Resource Training for SW Convergence", <http://www.hankyung.com/news/app/newsview.php?aid=2014052674831>, May 29, 2017.
- [21] SW oriented University Home page of Pusan University, SW Convergence Education, <https://www.software.kr/um/um02/um0207/um020712.do>, May 29, 2017.
- [22] SW oriented University Home page of Ajou University, SW Convergence Education, <https://www.software.kr/um/um02/um0207/um020708.do>, May 29, 2017.
- [23] SW oriented University Home page of Kyungpook University, SW Convergence Education, <https://www.software.kr/um/um02/um0207/um020703.do>, May 29, 2017.
- [24] SW oriented University Home page of Sungkyunkwan University, SW Convergence Education, <https://www.software.kr/um/um02/um0207/um020706.do>, May 29, 2017.
- [25] M. J. Oh, "Non-Major Students' Perceptions of Programming Education Using the Scratch Programming Language", The Journal of Korean Association of Computer Education, Vol. 20, No. 1, pp.1-11, 2017.

서 주 영(Seo, Joo Young)



- 2009년 2월 : 이화여자대학교 컴퓨터공학과(공학박사)
- 2009년 3월 ~ 2009년 8월 : 이화여자대학교 컴퓨터공학과 연구교수
- 2009년 9월 ~ 2016년 2월 : 아주대학교 소프트웨어학과 강의교수
- 2016년 3월 ~ 현재 : 아주대학교 다산학부대학 교수

- 관심분야 : 소프트웨어공학, SW 테스트, SW 융합교육
- E-Mail : jyseo@ajou.ac.kr