STUDENT SUCCESS MODEL IN PROGRAMMING COURSE: A CASE STUDY IN UUM

SALAM ABDULABBAS GHANIM

MASTER OF DEGREE
UNIVERSITI UTARA MALAYSIA
2014

STUDENT SUCCESS MODEL IN PROGRAMMING COURSE:
A CASE STUDY IN UUM

A thesis submitted to Dean of Awang Had Salleh Graduate School in

Partial Fulfillment of the Requirements for the Degree

Master of Science of Information Technology

University Utara Malaysia

By

Salam Abdulabbas Ghanim

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUMCollege of Arts and Sciences
Universiti Utara Malaysia

Abstrak

Kesukaran dan kerumitan dalam pengaturcaraan komputer telah dianggap sebagai punca kadar kegagalan dan keciciran yang tinggi. Pengaturcaraan telah dianggap oleh pelajar novis dan pertengahan, malah pelajar cemerlang juga sebagai satu kursus yang memerlukan kaedah pembelajaran yang pelbagai dengan menghasilkan dapatan yang pelbagai. Faktor-faktor kejayaan kursus pengaturcaraan di institusi pengajian tinggi telah dikaji. Rekod di Universiti Utara Malaysia (UUM) menunjukkan 38% dari pelajar semester satu ijazah sarjanamuda yang mengambil kursus pengaturcaraan dalam tahun 2013 telah gagal. Ini merupakan motivasi bagi kajian ini, yang meletakkan matlamat untuk mengenalpasti faktor praktikal yang mempengaruhi kejayaan dalam kursus pengaturcaraan, dan untuk menokok dapatan teoritikal di kalangan dapatan-dapatan sediaada oleh kajian lain. Kaedah kuantitatif telah digunakan, dengan mendapatkan data dari 282 responden yang telah disampelkan di kalangan pelajar sarjanamuda dan sarjana Teknologi Maklumat (IT) dan Teknologi Komunikasi dan Maklumat (ICT). Setelah data ditapis dan dibersihkan, dengan empat rekod yang mengandungi data terpencil dihapuskan dari senarai, ujian-T bebas, korelasi, dan regresi dijalankan bagi menguji hipotesis yang telah dibentuk. Dapatan dari Korelasi Pearson menunjukkan alatan pengajaran, konsep OOP, motivasi, penilaian kursus, dan keupayaan matematika mempunyai hubungan positif dengan pencapaian akademik. Manakala, ketakutan mempunyai hubungan yang negatif. Analisis regresi seterusnya menunjukkan hubungan adalah kuat, kecuali hubungan negatif iaitu ketakutan dengan pencapaian akademik. Ujian-T bebas pula membuktikan perbezaan antara kumpulan yang telah mempunyai pengalaman dan yang belum mempunyai pengalaman tidak wujud.

Keywords: Pengaturcaraan berasaskan objek, Java, kesukaran pengaturcaraan, pembelajaran, faktor

Abstract

The complexity and difficulty ascribed to computer programming has been asserted to be the causes of its high rate of failure record and attrition. It is opined that programming either to novice, middle learner, and the self-branded geeks is always a course to be apprehensive of different studies with varying findings. Studies on factors leading to the success of programming course in higher institution have been carried out. The record at Universiti Utara Malaysia (UUM) shows that 38% of semester one undergraduate students failed the programming course in 2013. This really motivates this study, which aims at investigating the practical factors affecting the success of programming courses, and to position its' theoretically findings to complement the existing findings. Data were gathered using a quantitative approach, in which a set of questionnaire were distributed to 282 sampled respondents, who are undergraduate and postgraduate students of Information Technology (IT) and Information and Communication Technology (ICT). Having screened and cleaned the data, which led to the deletion of four outlier records, independent T-test, correlation, and regression were run to test the hypotheses. The results of Pearson correlation test reveal that teaching tools, OOP concepts, motivation, course evaluation, and mathematical aptitude are positively related to academic success in programming course, while fear is found to be negatively related. In addition, the regression analysis explains that all the elicited independent variables except fear are strongly related. Besides, the independent T-test also discovers no deference between groups with and without previous programming experience.

Keywords: Object Oriented Programming, Java, programming difficulties, learning, Factors

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis.

Special appreciation goes to my supervisor, Mdm Alawiyah Abd Wahab, for her supervision and constant support. Her invaluable help of constructive comments and suggestions throughout the success of this research. This thesis would not have been possible without her help, support and her patience.

I sincerely thank to my evaluators Dr. Mazni Omar and Ms. Rohaida Romli, for graciously reviewing this work and giving valuable suggestion and comments on my work.

My deepest gratitude goes Prof. Dr. Huda (Dean, College of Arts and Sciences), Dr. Norliza , Dr. Hizbullah and all administrative staff of school of information technology specially Madam latifah.

I would also like to say a big thanks all UUM lecturers and staff members at the School of Computing who were kind enough to give me their precious time and assistance, without which I would not have been able to complete this Master's Thesis.

I am indebted and thankful to the Chancellor of University Utara Malaysia who referred me to valuable e-resources at the Sultanah Bahiyah Library.

Sincere thanks to all my friends especially Nasser Jabir, was always willing to help and give his best suggestions. I have been a lonely without him, and others for their kindness and moral support during my study. Thanks for the friendship and memories.

Last but not the least, I would like to thank my family: my Mother. She was always there praying me. Also to my elder brothers, and elder sisters. They were always supporting me and encouraging me with their best wishes. Special thank to my cousin Mohammed Tuama, Be on the go to do any requirements in my country when i ask.

Finally, I would like to thank my wife for her personal support and great patience at all times. She was always there stood by me through the good times and bad.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE
INTRODUCTION

1.0 Background of the Study

Modern curriculum needs to emphasize the development of programming skills for citizens of a technological society (Pejcinovic, Holtzman, Chrzanowska, & Jeske, 2013). Programming is a cognitive activity that requires abstract representations and logical expressions. The program must translate abstract representations into correct codes by using a formal language to create, modify, reuse, or debug a program (Wiedenbeck, 2005). Furthermore, programming is often viewed as a problem-solving activity rather than a linguistic activity, often ignoring the fact that programming languages are a case of formal languages. The interpretation of formal languages is unique for every individual.

Programming skills are an essential part of computer science (CS) and information technology (IT) courses (Raina Mason, Cooper, & Raadt, 2012). Robins, Rountree, and Rountree (2003a) argue that programming skills are useful in programming knowledge and strategies, such as program generation and comprehension. Programming can also lead to a rewarding career, such as an analyzer, programmer, or debugger.

Zdancewic and Weirich (2013) state that programming is a conceptual foundation in the study of computations. Programming is a prerequisite for almost every other course in CS. Renumol, Jayaprakash, and Janakiram (2009) said that "*programming is the process of writing, testing and debugging of computer programs using different programming languages*." However, according to

The contents of the thesis is for internal user only

REFERENCES

Alphonce, C., & Ventura, P. (2002). Object Orientation in CS1-CS2 by Design. *ACM SIGCSE Bulletin*, *34*(3). doi:10.1145/637610.544437.

Armoni, M., Gordon, M., & Harel, D. (2012). The Effect of Previous Programming Experience on the Learning of Scenario-Based Programming. *In Proceedings of the 12th Koli Calling International Conference on Computing Education Research.ACM*, 151–159.

Astrachan, & T. Selby, J. U. (2006). An object-oriented, apprenticeship approach to data structures using simulation. *In Frontiers in Education Conference, 1996. FIE'96. 26th Annual Conference., Proceedings of* (Vol. 1, pp. 130-134). IEEE.

Atieno, O. (2009). An Analysis of the Strengths and Limitations of Qualitative and Quantitative Research Paradigms. *Problems of Education in the 21st century, 13*, 13–18.

Babbie, E. (2010). *The Practice of Social Research,12th Edition.* Wadsworth Cengage Learning, USA.

Bailie, F. (2003). Objects First - Does It Work ? *Journal of Computing in Small Colleges*, *19*(2), 303–305.

Bennedsen, J., & Caspersen, M. E. (2006). Abstraction ability as an indicator of success for learning object-oriented programming? *ACM SIGCSE Bulletin, 38*(2), 39. doi:10.1145/1138403.1138430.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39*(2), 32. doi:10.1145/1272848.1272879.

Bennett, G., Fisher, M., & Lees, B. (2011). Object-Oriented Programming with Objective-c. *In Objective-C for Absolute Beginners*, 87–102.

Bergin, S., & Reilly, R. (2005a). Programming : Factors that Influence Success. *In ACM SIGCSE Bulletin*, 411–415.

Bergin, S., & Reilly, R. (2005b). The influence of motivation and comfort-level on learning to program. *In Proceedings of the PPIG*, (June), 293–304.

Biddle, R., & Tempero, E. (1998). Java Pitfalls for Beginners. *SIGCSE Bulletin*, *30*(2), 48–52.

Biju, S. M. (2013). Difficulties in understanding object oriented programming concepts. *In Innovations and Advances in Computer, Information, Systems Sciences, and Engineering (pp. 319-326).* Springer New York.

Black, A. P., Bruce, K. B., Homer, M., Noble, J., Yannow, R., Weishaupt, A., & Hazlitt, W. (2013). Seeking Grace : A new object-oriented language for novices. *In Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 129–134.

Blackwell, A. F. (1996). Metacognitive theories of visual programming: what do we think we are doing? *Proceedings IEEE Symposium on Visual Languages*, 240–246. doi:10.1109/VL.1996.545293.

Bougie, R., & Sekaran, U. (2010). *Research methods for business (5th ed.).* West Sussex, United Kingdom: John Wiley & Sons Ltd.

Bruce, C., Buckingham, L., Hynd, J., Mcmahon, C., Roggenkamp, M., & Stoodley, I. (2004). Ways of Experiencing the Act of Learning to Program : A Phenomenographic Study of Introductory Programming Students at University. *Journal of Information Technology Education, 3, 143-160.* Retrieved from http://www.jite.org/documents/Vol3/v3p143-160-121.pdf on 3rd February, 2014.

Bruce, K. B. (2005). Controversy on how to teach CS 1: A discussion on the SIGCSE-members mailing list. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, *37*(2), 111–117.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and Information Technologies*, *83*, 65–83.

Bryman, A. (2004). *Social research methods.* New York: Oxford University Press Inc.

Burkhardt, J., Détienne, F., Wiedenbeck, S., Voluceau, D. De, & Chesnay, L. (1997). Mental Representations Constructed by Experts and Novices in Object-Oriented Program Comprehension. *In Human-Computer Interaction INTERACT'97, pp. 339-346).* Springer US., 339–346.

Burns, R. B. (1997). *"Introduction to Research Methods in Education,"* Melbourne, Victoria: Longman Cheshire.

Burton, P. J., & Bruhn, R. E. (2003). Teaching Programming in the OOP, *ERA*, *35*(2), 111–114.

Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. *In Proceedings of Ascilite*, 99–107.

Byrne, M. D., Catrambone, R., & Stasko, J. T. (1999). Evaluating animations as student aids in learning computer algorithms. *Computers & Education*, *33*(4), 253–278. doi:10.1016/S0360-1315(99)00023-8.

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '01*, 49–52. doi:10.1145/377435.377467.

Carbone, A., & Hurst, J. (2009). An E xploration of I nternal F actors I nfluencing S tudent L earning of P rogramming. *Proceedings of the Eleventh Australasian Conference on Computing Education*, *95*(Ace), 25–34.

Carbone, A., Hurst, J., Mitchell, I., & Gunstone, D. (2009). An exploration of internal factors influencing student learning of programming. *In Proceedings of the Eleventh Australasian Conference on Computing Education . Australian Computer Society, Inc.., 95*, 25–34.

Carlisle. (2009). Raptor: A Visual Programming Environment For Teaching Object-Oriented Programming*. *Journal of Computing Sciences in Colleges, 24,* 275–281.

Caspersen, M. E., Kölling, M., Ct, K., & Beck, K. (2006). A Novice ' s Process of Object-Oriented Programming. *Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and applications,ACM*, 892–900.

Cheney, E. E. W., & Kincaid, D. R. (2012). *Numerical mathematics and computing. C*engage Learning.

Clark, D., MacNish, C., & Royle, G. F. (1998). Java as a teaching language—opportunities, pitfalls and solutions. *In Proceedings of the 3rd Australasian Conference on Computer Science education,ACM*, 173–179.

Close, D. Kopec, and J. A. (2000). CS1: Perspectives on Programming Languages and the Breadth-First Approach. *In Proceedings of the 5th Annual CCSC Northeastern Conference on Computing in Small Colleges*, 1–7.

Cooper, S., Dann, W., & Pausch, R. (2003). Teaching objects-first in introductory computer science. *ACM SIGCSE Bulletin*, *35*(1), 191. doi:10.1145/792548.611966

Costa, C. J., Iscte, A., & Pierce, R. (2009). Evaluating Information Sources for Computer Programming Learning and Problem Solving. *In Proceedings of the 9th WSEAS International Conference on APPLIED COMPUTER SCIENCE,* 218–223.

Crawford, S. & Boese, E. (2006). ActionScript: a gentle introduction to programming. *Journal of Computing Sciences in Colleges*, *21*(3), 156–168.

Dehnadi, S., & Bornat, R. (2006). *The Camel Has Two Humps* (working title). Middlesex University, UK. 1–21.

Eckerdal, A., Box, P. O., & Thun, M. (2005). Novice Java programmers' conceptions of object and class, and variation theory. *In ACM SIGCSE Bulletin* (Vol. 37, No. 3, pp. 89-93).

Eckerdal. (2006). Novice Students ' Learning of Object-Oriented Programming.

El-Zakhem, I., & Melki, A. (2013). Difficulities In Learning Programming LanguagesAmong Freshman Students. *INTED2013 Proceedings*, 1202–1206.

Entwistle, N. (1998). Motivation and approaches to learning: motivation and conceptions of teaching. *In: Brown, S., Armstrong, S., Thompson, G. (Eds.), Motivating Students. Kogan Page, London, United Kingdom.*

Floyd, B., & London, R. (1970). I . Notes on Structured Programming: *Technological University Eindhoven.*

Garner, S. K. (2001). Cognitive load reduction in problem solving domains.

Garner, S., Haden, P., & Robins, A. (2005). My Program is Correct But it Doesn ' t Run: A Preliminary Investigation of Novice Programmers ' Problems. *In Proceedings of the 7th Australasian Conference on Computing Education, 42,* 173–180.

Garrido, J. M. (2004). Object-Oriented Programming: From Problem Solving to Java. *Firewall Media.*

Georgatos (2008). How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view?. *arXiv preprint arXiv:0809.1437.*

Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. *In International Conference on Engineering Education–ICEE.*

Graf, S., Lan, C. H., & Liu, T.-C. (2009). Investigations about the Effects and Effectiveness of Adaptivity for Students with Different Learning Styles. *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, 415–419. doi:10.1109/ICALT.2009.135

Gries, D. (1974). What should we teach in an introductory programming course? *SIGCSE '74: Proceedings of the Fourth SIGCSE Technical Symposium on Computer Science Education*.

Grinnell, R. jr. (ed.). (1993). *Social Work, Research and Evaluation", (*4th ed), Illinois, F.E Peacock Publishers.

Grogono. (1989). Comments, assertions and pragmas. *ACM SIGPLAN Notices,*, *24*(3), 79–84.

Gross, P. & Powers, K. (2005). Evaluating assessments of novice programming environments. *Proceedings of the 2005 International Workshop on Computing Education Research ICER '05*, 99–110.

Guthrie, R., Yakura, E., & Soe, L. (2011). How Did Mathematics and Accounting Get So Many Women Majors ? What Can IT Disciplines Learn?, *1886*(909), 15–19.

Hadjerrouit, S. (1998). Java as First Programming Language: A Critical Evaluation, *30*(2).

Hadjerrouit, S. (1998). Java as First Programming Language: A Critical Evaluation,. *ACM SIGCSE Bulletin*, *30*(2), 43–47.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E. &Tattam, R. L. (2010). *Multivariate Data Analysis* (7th ed.), New Jersey: Pearson Education Inc.

Hardy, C., Heeler, P., & Brooks, D. (2006). Are High School Graduates Technologically Ready For Post-Secondary Education? *Journal of Computing Sciences in Colleges*, *21*(4), 52–60.

Hasan, N. M. and Y. (2001). Challenges in teaching java technology. *Informing Sci.*, 365–371.

Havenga, H. M. (2006). *An Investigation Of Students ' Knowledge , Skills And Strategies During Problem Solving In Object-Oriented Programming.*

Havenga, M., Mentz, E., & De Villiers, R. (2008). Knowledge, skills and strategies for successful object-oriented programming: a proposed learning repertoire. *South African Computer Journal*, *42*, 1–8.

Helme & Clarke. (2009). Identifying cognitive engagement in the mathematics classroom. *Mathematics Education Research Journal*, *13*, 133–153.

Helme, S. U. E., & Clarke, D. (2001). Identifying cognitive engagement in the mathematics classroom. *Mathematics Education Research Journal*, 131–153.

Henderson, R., & Zorn, B. (1994). A Comparison of Object-oriented Programming in Four Modern Languages, *24*(June), 1077–1095.

Herman, N. S., & Salam, S. B. (2011). A Study of Tracing and Writing Performance of Novice. *In Software Engineering and Computer Systems . Springer Berlin Heidelberg.*, 557–570.

Holden, Ronald B. (2010). "Face validity". In Weiner, Irving B.; Craighead, W. Edward. *The Corsini Encyclopedia of Psychology* (4th ed.). Hoboken, NJ: Wiley. pp. 637–638. ISBN 978-0-470-17024-3.

Holland, S., Griffiths, R., Woodman, M., Hall, W., Keynes, M., & Kingdom, U. (1997). Avoiding Object Misconceptions. *In ACM SIGCSE Bulletin*, *29*, 131–134.

Hossein, S. (2007). *Response modeling in direct marketing. Master thesis, Department of business administration and social science,* University of Technology, Iran.

Howles, T. (2007). Preliminary Results Of A Longitudinal Study Of Computer Science Student Trends, Behaviors and Preferences. *Journal of Computing Sciences in Colleges*, *22*, 18–27.

Jenkins, T. (2001a). Teaching programming–A journey from teacher to motivator. *Paper Presented at the The 2nd Annual Conference of the LSTN*, Center for Information and Computer Science.

Jenkins, T. (2001b). The motivation of students of programming. *ACM SIGCSE Bulletin*, *33*(3), 53–56.

Jenkins, T. (2002). On the difficulty of learning to program. *In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 53–58.

Kaplan, R., & Sacuzzo, D. (2008). *Psychological Testing: Principles, applications and Issues, Brooks Cole. Pacific Grove, CA.*

Kerlinger, F. N. (1986). *Foundation of Behavioral Research, 3th edition. New York: Holt, Rinehart & Winston*.

Kidder, L. H., & Judd, C. M. (1986). *Research methods in social relations (5th ed.). New York: Holt, Rinehart and Winston*.

Kimberly A. Barchard. (2003). Does Emotional Intelligence Assist in the Prediction of Academic Success? *Educational and Psychological Measurement*; 6(3), 840-858.

Kölling, M. (1999). The problem of teaching object-oriented programming. *In ACM Sigplan Notices*, *11*(8), 8–15.

Kölling. (2003). The BlueJ system and its pedagogy 1. *Computer Science Education*, *13*(4),1–12.

Kumar, R. (2011). *Research methodology: A step-by-step guide for beginners (3rd ed.).*Thousand Oaks, CA: SagePublications Inc.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3), 14. doi:10.1145/1151954.1067453

Lakshman, M., Sinha, L., Biswas, M., Charles, M., & Arora, N. K. (2000). Quantitative Vs qualitative research methods. *The Indian Journal of Pediatrics*, *67*(5), 369–377. doi:10.1007/BF02820690

Li, T., Liu, W., Mao, X., & Zhou, H. (2013). Introduction to Programming : Science or Art? *In Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 4503.

Macklem, Gayle L. (1990). Measuring aptitude. *Practical Assessment, Research & Evaluation*, 2(5).

Madden, M., & Chambers, D. (2002a). Evaluation of Student Attitudes to Learning the Java Language. *In Proceedings of the inaugural conference on the Principles and Practice of programming, 2002 and Proceedings of the second workshop on Intermediate representation engineering for virtual machines,* (pp. 125-130). National University of Ireland.

Madsen, O. L., & Møller-Pedersen, B. (1988). What object-oriented programming may be-and what it does not have to be. *In ECOOP'88 European Conference on Object-Oriented Programming (pp. 1-20). Springer Berlin Heidelberg.*

Mason, R. (2012). Designing introductory programming courses: the role of cognitive load.

Mason, R., Cooper, G., & Raadt, M. De. (2012). Trends in Introductory Programming Courses in Australian Universities – Languages , Environments and Pedagogy. *Computing Education Conference (ACE2012), 123,* 33–42.

Matravers, J. (2011). Introduction to computer systems architecture and programming. *University of London*.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B., & Al., E. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, *33*(4), 125–180.

Mcgettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., & Mander, K. (2005). Grand Challenges Education. *The Computer Journal*, *48*(1), 49–52.

McGettrick. (2005). Grand challenges in computing: Education--A summary. *The Computer Journal,*, *48*(1), 42–48.

Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C. St., & Thomas, L. (2006). A cognitive approach to identifying measurable milestones for programming skill acquisition. *In ACM SIGCSE Bulletin*, *38*, 182–194. doi:10.1145/1189215.1189185

Mills, H. D. (1972). *Mathematical Foundations for Structured Programming*.

Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming — Views of Students and Tutors. *Education and Information Technologies*, 55–66.

Moderator, O. A., Koffman, E., Kölling, M., & Reges, S. (2005). Resolved : Objects Early Has Failed. *In ACM SIGCSE Bulletin*, *37*, 451–452.

Mody, R. P. (1991). C in education and software engineering. *ACM SIGCSE Bulletin*, *23*(3), 45–56. doi:10.1145/126459.126471

Mow, I. T. C. (2008). Issues and difficulties in teaching novice computer programming. *In Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education* (pp. 199-204). *Springer Netherlands.*

Nikishkov, G. P., Nikishkov, Y. G., & Savchenko, V. V. (2003). Comparison of C and Java Performance in Finite Element Computations. *Computers & Structures*, *81*(24), 2401–2408.

Notani, A. S. (1998). Moderators of perceived behavioural control's predictiveness in the theory of planned behaviour: A meta-analysis. *Journal of Consumer Psychology*, *7*(3), 247–271.

Object Management Group. (2003). Unified Modeling Language Specification. *Version 1.5 March 2003, Doc. Number formal/03-03-01*.

Pair, C. (1993). Programming, programming languages and programming methods. *Psychology of Programming*, 9-19.

Pallant, J. (2003). *SPSS survival manual: A step by step guide to data analysis using SPSS for Windows (Version 10)*. Australia: Allen & Unwin.

Pallant, J. (2011). *For the SPSS Survival Manual website* , go to www.allenandunwin.com/spss This is what readers from around the world say about the SPSS Survival Manual.

Patil, P. S. P. (2009). The effect of developments in student attributes on success in Programming of management students. *In Education Technology and Computer, 2009. ICETC'09. International Conference on IEEE.*, 191–193. doi:10.1109/ICETC.2009.35

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Uni, J. M., … Paterson, J. (2007). A Survey of Literature on the Teaching of Introductory Programming. *In ACM SIGCSE Bulletin*, *39*(4), 204–223.

Pejcinovic, Holtzman, M., Chrzanowska-Jeske, & W. (2013). Just because we teach it does not mean they use it: Case of programming skills. *In Frontiers in Education Conference*, 1287–1289.

Poo, D. C., Kiong, D. B. K., & Ashok, S. (2007). *Object-Oriented Programming and Java* (2nd Editio). Springer.

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K. J., Proulx, V. & Carlisle, M. (2006). Tools for teaching introductory programming: what works? *ACM SIGCSE Bulletin , Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education SIGCSE '06*, *38*(1), 560–561.

Quenemoen, R., Thompson, S. & Thurlow, M. (2003). *Measuring academic achievement of students with significant cognitive disabilities: Building understanding of alternate assessment scoring criteria*(Synthesis Report 50). Minneapolis, MN: University of Minnesota, National Center on Educational Outcomes.

Reis, C., Tx, H., & Cartwright, R. (2004). Taming a Professional IDE for the Classroom. *In ACM SIGCSE Bulletin*, *36*, 156–160.

Renumol, V. G., Jayaprakash, S., & Janakiram, D. (2009). Classification of Cognitive Difficulties of Students to Learn Computer Programming. *Indian Institute of Technology, India.*

Rinard, M. C., Scales, D. J., & Lam, M. S. (1993). Jade: A High-Level , *Language for Parallel Programming,26*, 28–28.

Robins, A., Rountree, J., & Rountree, N. (2003a). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172. doi:10.1076/csed.13.2.137.14200

Robins, A., Rountree, J., & Rountree, N. (2003b). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172. doi:10.1076/csed.13.2.137.14200

Rogerson, C., & Scott, E. (2010). The Fear Factor: How It Affects Students Learning to Program in a Tertiary Environment. *Journal of Information Technology Education*.

Rosson, M. B., & Alpert, S. R. (1990). The cognitive consequences of object-oriented design. *Human-Computer Interaction*, *5*(4), 345–379.

Rowe, I. M. and G. (2002). Difficulties in learning and teaching programming-Views of students and tutors. *J. Edu. & Info. Tech.*, *7,* 55–66.

Sajaniemi, J., & Kuittinen, M. (2003). Program animation based on the roles of variables. *Proceedings of the 2003 ACM Symposium on Software Visualization - SoftVis '03*, 7. doi:10.1145/774834.774835

Satzinger, J.W. & Ørvik, T. U. (2001). *The Object-Oriented Approach. Concepts, System Development, and Modeling with UML.*

Saunders, M., Lewis, P., & Thronhill, A. (2003). *Research method for business students (3 ed). England:* Person Education Limited.

Schach, S. R. (2005). *Object-Oriented and Classical Software Engineering. (6th ed.). Boston: McGraw-Hill.*

Schreiner, w. (2011). *Introduction to Programming* (pp. 1–156).

Schulte, C., & Bennedsen, J. (2006). What do teachers teach in introductory programming? *In Proceedings of the Second International Workshop on Computing Education Research ACM*, p 17–28.

Sebesta, R. W. (2004). *Concepts of Programming Languages. (6th ed.).* Boston: Pearson Addison Wesley.

Sekaran, U. (2003). *Research Methods for Business: A skill building approach.* John Wiley and Sons Inc., New York.

Sekaran, U., & Bougie, R. (2009). *Research methods for business: A skill building approach.* Wiley: London.

Sekaran, U., & Bougie, R. (2010). *Research methods for business: A skill building approach.* Wiley.

Sharp & Schultz. (2013). An Exploratory Study of the use of Video as an Instructional Tool in an Introductory C# Programming Course. *Information Systems Education, 11*(6).

Sicilia, M.-Á. (2006). Strategies for teaching object-oriented concepts with Java. *Computer Science Education, 16*(1), 1–18. doi:10.1080/08993400500344431

Simon, S., Fincher, S., Robins, A., Baker, B, Box, I., Cutts, Q., de Raadt., M., Haden, P., Hamer, J., H., & M., Lister, R., Petre, M., Sutton, K., Tolhurst, D., & Tutty, J. (2006). Predictors of Success in a First Programming Course. *In Proceedings of the 8th Australasian Conference on Computing Education*, *52*, 189–196.

Singer, Wing Hang Li , David R. White, J. (2013). JVM-Hosted Languages: They Talk the Talk, but do they Walk the Walk? *In Proceedings of the 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools,ACM*, 101–112.

Sivasakthi, M., & Rajendran, R. (2011a). Learning difficulties of " object-oriented programming paradigm using Java ": students ' perspective. *Indian Journal of Science and Technology*, *4*(8), 983–985.

Sivasakthi, M., & Rajendran, R. (2011b). Learning difficulties of " object-oriented programming paradigm using Java ": students ' perspective. *Indian Journal of Science and Technology*, *4*(8), 983–985.

Smith, H. W. (1991). *Strategies of social research:* Holt, Rinehart and Winston.

Smith. (2012). *Research Methodology: A Step-by-step Guide for Beginners.*

Soloway, S. and. (1989). *Studying the novice programmer*. *H*illsdale, New Jersey, United States: Lawrence Erlbaum.

Souza, D. D., Hamilton, M., Thevathayan, C., Harland, J., Walker, C., & Muir, P. (2008). Transforming Learning of Programming: A Mentoring Project. *In Proceedings of the Tenth Conference on Australasian Computing Education*, *78*, 78–84.

Stroustrup, B. (1991). What is "Object-Oriented Programming"?(1991 revised version).

Tabachink, B. G. &Fidell, L. S. (2006). *Using Multivariate Statistics* (5th Ed.), USA: Pearson Education Inc.

Thompson, P. (1995). *Constructivism in education* (p. 159). Hillsdale, NJ: Lawrence Erlbaum.

Turner, J. A., & Zachary, J. L. (2001). Javiva: A Tool for Visualizing and Validating Student-Written Java Programs. *In ACM SIGCSE Bulleti*, *33*(1), 45–49.

Ulloa, M. (1980). Teaching and learning computer programming: a survey of student problems, teaching methods, and automated instructional tools. *ACM SIGCSE Bulletin*, *12*(2), 48–64.

Von Wangenheim, C. G., & Shull, F. (2009). voice of evidence. *IEEE, 26*(2), 92–94.

VRajaraman. (1998). Programming Languages. *http://ezproxy.unimap.edu.my:Comparison and Classification of Programming Languages (Springer)*, 1–12.

Weisfeld, M. (2004). *The Object-Oriented Thought Process. (2nd ed.).Developer's Library*.

Weiss, M. A. (2000). Data structures and problem solving using Java. *ACM SIGACT News*, *29*(2), 42–49. doi:10.1145/288079.288084

Wenger, E. (1998). Communities of practice: Learning as A social practice. *The Systems Thinker*, *9*(5).

Wiedenbeck, S. (2005). Factors affecting the success of non-majors in learning to program. *Proceedings of the 2005 International Workshop on Computing Education Research - ICER '05*, 13–24. doi:10.1145/1089786.1089788

Wiedenbeck, S. et al. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers, 11*, 255–282.

Wiedenbeck, S., & Labelle, D. (2004). Factors Affecting Course Outcomes in Introductory Programming, (April), 97–110.

Williams, K. C., & Williams, C. C. (2011). Five key ingredients for improving student motivation. *Research in Higher Education Journal*, 1–23.

Wilson, B. C., & Shrock, S. (2001). Contributing to Success in an Introductory Computer Science Course : A Study of Twelve Factors. *In ACM SIGCSE Bulletin*, *33*, 184–188.

Wit, K. De, Heerwegh, D., & Verhoeven, J. C. (2012). Do ICT Competences Support Educational Attainment at University? *Journal of Information Technology Education: Research 11. Available at http://www.jite.org/documents/Vol11/JITEv11p001-025DeWit1037.pdf [Accessed 22-05-2012]*, *11*.

Xinogalos, S., Sartatzemi, M., & Dagdilelis, V. (2006). Studying Students ' Difficulties In An Oop Course Based On Bluej. *In IASTED International Conference on Computers and Advanced Technology in Education*, 82–87.

Yau, J. Y., & Joy, M. (2004). Introducing Java: the Case for Fundamentals-first.

Zdancewic& Weirich. (2013). *Programming Languages and Techniques.*University of Pennsylvania, 1–387.

Zikmund, W., Babin, B., Carr, J., & Griffin, M. (2010). *Business research methods South*-Western Cengage: Canada.