Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 2 Issue 12, December - 2016

# Girls Who Code: Assessing Programming Language Preferences Among Female Undergraduate CS Students

**Francisca O. Oladipo**, PhD, FASI, Citp
Computer Science Department, Faculty of Science
Federal University Lokoja
Lokoja, Nigeria
francisca.oladipo@fulokoja.edu.ng

***Abstract*— This paper reports on a study of the programming language preferences among some female Computer Science undergraduate students in four Nigerian higher institutions. The research is aimed at gaining an insight into the factors that influence the choice of programming language among female CS undergraduates as well as determine what effects the choice of programming language has on coding efficiencies and design thinking/problem solving abilities of the students. Questionnaires were administered to a total of 35 female students of computer science from four tertiary institutions in north-central Nigeria. The researcher defined two broad classes of assessment criteria based on personal interests and the technical properties of the language for the study, and the results showed the students were indifferent to technical features like biding time for variables and dynamic debugging, but will show preference for languages that provide support for modularity and components re-use, as well as those that are easy to learn and understand. The respondents revealed that they have personal favourites and that they would prefer a language that is emphasized in the curriculum and taught to them in addition to considering the complexity of the language elements, target platforms and suitability for the problem at hand. After the return of the questionnaires, onsite interviews were conducted for 15 female respondents who were indifferent on the technical consideration and exerts from their responses are illustrated in a dynamic jMap.**

***Keywords*— *diversity; programming; logical thinking; language preference; semantic dimensions***

## I. INTRODUCTION

Coding or programming is defined as the translation of an algorithm into the syntax and semantics of a programming language in order for it to be executed by the computer [1]. The study of the art and languages of programming is a major component of undergraduate computer science curriculum as specified in the benchmark by the general regulatory body for university education in Nigeria, the National Universities Commission (NUC) and the Computer Professionals Registration Council of Nigeria (CPN) which specifically regulates the practice of the computing profession in Nigeria.

***"Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains***. " -Bill Gates.

Programming is considered as one of the most important aspect of computing as it enables logical thinking in computer scientists, bring their solutions to fruition and enables them build reliable software systems. This one of the reasons why for computer scientists, emphasis is more on learning and using computer programming languages than using applications as tools. It is the position of [2] that increased problem solving skills and computational thinking are some of the benefits of coding in a programming language. Other benefits include: providing a better background to choosing and learning new languages as well as support for already known ones, Increased capacity to express ideas, as well as providing an overall advancement of computing [3].

As attractive and as important programming is, may female computer science majors still shy away from the art of learning programming languages. Several factors had been identified as being responsible for the low female interest in programming and programming languages. According to [4], genetically, females though are more intuitive and perceptive, the do not want to code as they are not drawn into the same interests as the male gender. A satirical publication by [5] identified an existing bias against applications developed by women as one of the reasons for less female enrolment in programming classes and consequently taking on tech jobs that involve using knowledge of programming languages.

This paper reports the result of a study conducted on female CS students in four tertiary institutions in Nigeria. The research is aimed at determining what factors influence their choice of programming languages and make recommendations to faculties based on these factors. The rest of the paper is organized as follows: a review of related work is conducted in the next section, which is followed by a description of the research instruments and methods.

A discussion of the obtained results follows and finally a concluding remark.

## II. Related Work

Very few studies exist to determine programming languages preferences among female learners in tertiary institutions. This section examines some research works that try to x-ray the criteria for selecting programming languages including an evaluation of the level of participation of students in computing generally and programming in particular.

Reference [6] conducted a research to identify and formalize the criteria used by professors in selecting introductory computer programming languages to teach students. The study developed an extensible set of language selection criteria, weighed each one, built a list of programming languages as nominated by the teaching faculties. A pilot study to test the validity of the language selection criteria was conducted and the model was refined as a result of the feedback from the pilot study.

In the same vein, an objective comparison of common programming languages based on decisions by teaching language creators was carried out by [7]. The researchers got prominent creators to generated a list of criteria, provide justification for them and these criteria are then used for choosing languages to be used for teaching to introductory programming classes. Reference [8] on the other hand proposed and developed a software teaching tool for a first course in programming languages. The resulting tool was tested by deploying it in teaching an introductory course in programming and the result showed the effectiveness of the tool in driving students' performance improvements in learning the programming concepts and reducing the time spent covering syntax, provided support for flexible switching from one language to another, bridged the gap between students from various backgrounds and led to increased enthusiasm in programming among students.

In response to the concerns about low participation of female in computing by some Australian universities, [9] conducted an exploratory research on the factors preventing female participation in computing courses and offered some suggestions towards increasing the female enrolment in computing courses in schools. Similarly, a preliminary experiment to measure the impact of students' programming assignment and their value on diversity was conducted by [10]. The programming assignment was entry level and evidence obtained from the result supported the notion that the assignment acted as a tool for students to lean to value diversity.

A study to identify the motivational factors in programming among engineering students in TATI University College was conducted by [11]. The researchers using qualitative and quantitative survey methods, measured students' perception and expectation in learning a new programming language. The study found two sources of motivation: intrinsic factors which are individual attitude and expectation and extrinsic factors, which are clear direction, reward and recognition, punishment and social pressure and competition. The researchers also suggested ways to improve programming languages learning in low-motivated students.

Establishing the relationship between different learning styles and learning performances in online studying environments was the focus of a research by [12]. The research was conducted among 62 sophomore students who enrolled in an online introductory programming course and using the Turkish version of Kolb's Learning Style Inventory (LSI) to measure the students' study habits, a significant relationship was established between learning styles, study habits, and learning performances.

A 2015 review by [13] enumerated some reasons why learning programming is important. It is the position of the author that in addition to sharpening intellect and developing logical reasoning, programming allows users to be "part of the creators' of the products they use, as well as gain an understanding of the logic and science behind them. The author further stated that since the ability to code separated those who conceive ideas to those who take them further by bringing them to life, the later categories of learners certainly will feel confident as (code) designers and (code) builders.

The entire world is being digitized and building software is being integrated into everyday life. Programming should be seen by girls as an art which enables communications and provide a gateway to constructing solutions to everything in life from fashion to medicine to space science. Programming creates a world of endless possibilities.

## III. Materials and Methods

### *A. Research Sample Space and Data Gathering*

This research was conducted among female undergraduate computer science students in four tertiary institutions in North-Central Nigeria. There were no special criteria for selecting these four institutions other than the fact of their proximity to the researcher. In addition, due to the few number of female students in the sciences, the only way to get a substantial sample space is by expanding the study sites beyond the researcher's present institution. The researcher also felt it is necessary to obtain diverse opinions across more than one institution. Primary data used in this study were obtained by distributing questionnaires to volunteers across the four institutions and conducting onsite oral interviews for some selected female third and fourth year student who had taken advanced programming courses. A total of 35 questionnaires were distributed to respondents and immediately collected back upon completion leading to a response rate of 100%. 15 onsite interviews were conducted and the interviewees' response to the touchpoints were recorded for building the pupil programmer's journey map.

In addition to the specific questions testing for factors responsible for programming language preferences, personal information like age, tribe,

course of study (though already known), reason for choosing course of study and sex (also already known) were asked in the survey. In order to measure the language preference, the research used a 5-point scale of response choices: Strongly Agree (5), Agree (4), Neutral (3), Disagree (2), Strongly Disagree (1).

### *B. Evaluation Criteria*

In order to determine what factors influenced the choice of programming languages among female CS undergraduates, the researcher developed a set of evaluation criteria. These were based on the informational criteria of programming languages knowledge and the sematic dimensions of the codes and coding process, the syntax and semantics of the language elements as well as other technical considerations. The survey questions were derived from these criteria.

#### *1) Informational Criteria: Personal interest test*

These set of questions were designed to test for students' preferences based on their personal interests which may be due to what they already know about language(s). The knowledge is not restricted to languages taught as part of the curriculum but also included those the students choose to learn on their own or are generally familiar with as a result of their association with the community of user. Questions were designed to test how many of the students would prefer a particular language based on the following informational criteria:

- Programming skills enforced in my department
- Simple language elements
- Ease of learning and understanding of the language
- Personal favorite
- Integration of static analysis tools
- Suitability for purpose
- Availability of support and user/developer community

#### *2) Semantic Dimensions*

The questions designed under the semantic dimensions tested the students' preferences based on the technical properties of the language environments and those of the codes and other by-products resulting from the language.

- ❖ The platform where the resulting application will run
- ❖ Support for modularity
- ❖ Support for component re-use
- ❖ Portability of the code developed in the language
- ❖ Support for dynamic debugging
- ❖ Static and dynamic typing
- ❖ Variable biding

The following graphic visualization tools were deployed to show the results of the survey and highlight the touchpoints:

1. Placeholder images
2. Static and dynamic journey maps

## IV. Discussion of Results

1. Percentage Distribution of the Respondent according to Sex

   All the respondents are female (Fig 1)

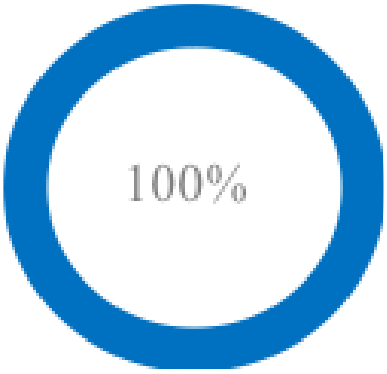


Fig. 1. *Demographics of the respondents.*

2. Percentage Distribution of the Oral Interviewees

   50% of the students who were neutral on preference based on technical factors were randomly selected for further discussions onsite in order to obtain deeper insight into the technical barriers (Figure 2).

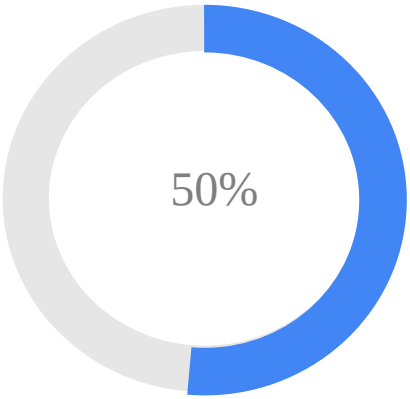


Fig. 2. *Percentage of respondents selected for onsite interview.*

The research recorded a response rate of 100% (Fig 1) as all the distributed questionnaires were immediately collected back upon completion. The findings according to the response to the questions are as follows:

### *A. Personal Interest Test*

The response to the informational criteria is summarized below (Table 1).

TABLE I. PREFERENCE BASED ON INFORMATIONAL CRITERIA

| Criteria | Preference (x/n; n=35) | | | | |
|---|---|---|---|---|---|
| | ***SA*** | ***A*** | ***N*** | ***D*** | ***SD*** |
| Programming skills enforcing in my department | 22 | 13 | - | - | - |
| Simple programming elements | 27 | 8 | - | - | - |
| Ease of learning and understanding | 30 | 5 | - | - | - |
| Personal favorite | 35 | - | - | - | - |
| Integration of static analysis tools | 2 | - | 30 | 3 | - |
| Suitability for purpose | 28 | 6 | 1 | - | - |
| Availability of support and user/developer community | 22 | 7 | 3 | 2 | 1 |

As seen in the data collected from the students (Table 1), one major factor that influence their language preference is the curriculum. All the respondent agree that they would demonstrate preference for those programming languages that are taught as part of the curriculum while making choices. 90% of the students prefer a language with simple and predictable language elements. Thus support for non-ambiguity, pre-defined operators' precedence levels and implied null initialization of variables are some factors that dictate language preferences. 85.71% of the students will strongly prefer an easy to learn language, as this saves time to development of their applications. All the students admit they would show preference for a programming language that they favour personally over any other language if they have a choice. 6.67% of the students would care about integration of static analysis tools as part of a programming language Integrated Development Environment (IDE), however, it was revealed during the onsite interview that the students had no idea what static analysis meant and were amazed to realize that it actually is being provided for in some of the popular IDEs they had worked with.

About 93% of the students would generally learn towards a language based on its suitability for the type of application being developed. Many of the students when interviewed orally responded that they usually search for a 'most appropriate' tool at the early stage of the programming problem solving in order chose a widely recommended (as suitable) language. Availability of support community and developer support system is another factor which the students considered as crucial to offering preference for a programming language. 71.42% of the students would strongly favour a programming environment that offers an avenue for developers using such tools to discuss solutions, get valuable advice when they experience difficulties, obtain information on new fixes and release, fork existing codes from code bases, etc.

TABLE II. PREFERENCE BASED ON SEMANTIC DIMENSIONS

| Criteria | Preference (x/n; n=35) | | | | |
|---|---|---|---|---|---|
| | ***SA*** | ***A*** | ***N*** | ***D*** | ***SD*** |
| Platform considerations | 24 | 11 | - | - | - |
| Modularity | 23 | 8 | - | 2 | 2 |
| Component re-use | 35 | - | - | - | - |
| Code portability | 35 | - | - | - | - |
| Dynamic debugging | 20 | 12 | 1 | - | 2 |
| Typing (static vs dynamic) | 12 | 21 | 1 | 1 | - |
| Biding (runtime vs compile time) | 2 | 7 | 23 | 2 | 1 |

Analyzing the research result showed that over 68% of the students would strongly consider the platform environment where the resultant application will run before choosing a programming language for development and considerations will be given to the target operating system(s) as well as the database systems and other middleware. While a mere 11% of the students would prefer a language whose syntax and semantics provide straight-forward support for modularization, 100% of them would show preference for a language that allows them re-use components (modules, functions, classes, blocks, etc). Equally, same number of students would adopt a language not constrained by hardware in any form or those that have standardized language and environment. Greater than 65% of the students wouldn't care about variable biding time while choosing what programming environment to deploy, but nearly all of them would consider whether variable typing is done at run time or compile time.

## V. CONLUSION

The researcher conducted a survey of factors that influence choices of programming language among diversity (female) undergraduate students of four tertiary institutions in Nigeria and reported the responses herein. Findings revealed that students demonstrated preferences for a language with large community following and support group, supports re-use of previously designed component, not dependent on hardware, easy to learn and contains non-complex language elements. It was also discovered that students generally have personal favourites among programming languages and would consider these over other technical considerations like target platform, biding and debugging. Exerts from the respondents'

onsite interview are illustrated in a dynamic jMap (Fig 3).

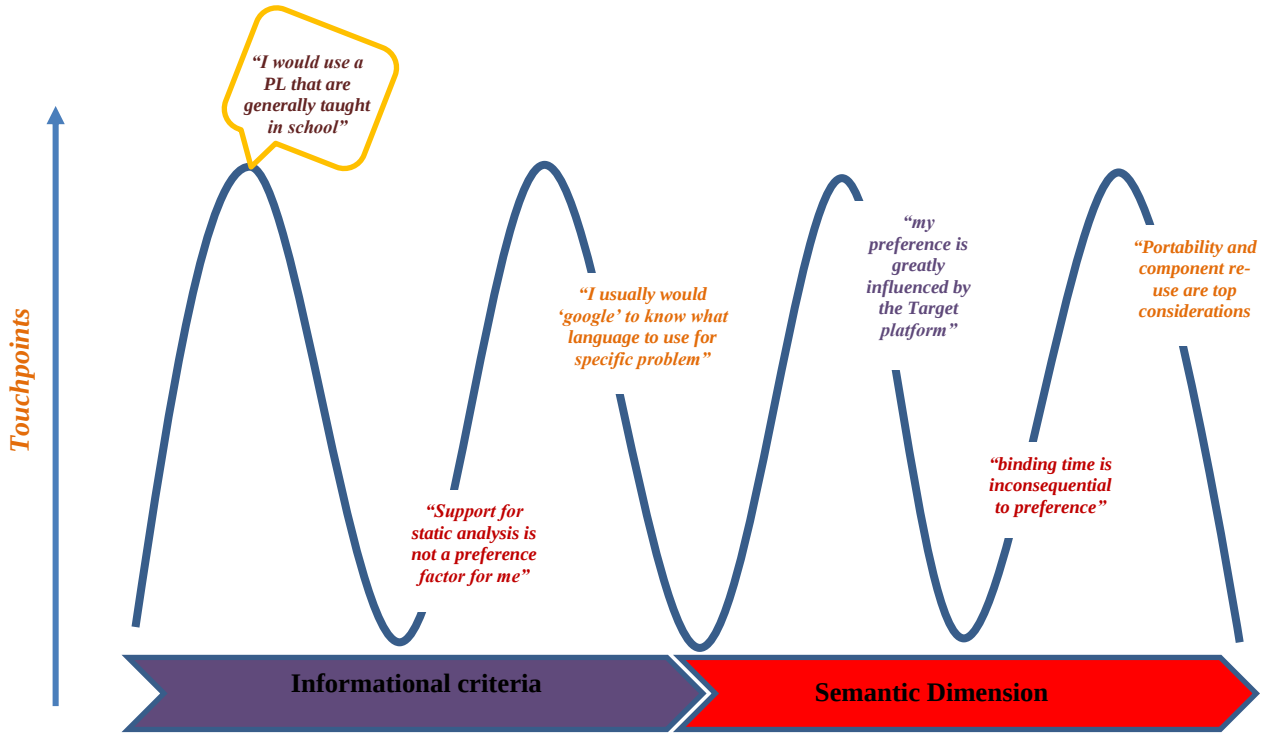


Fig. 3. *Dynamic jMap of students' onsite interview touchpoints.*

The author concludes that since the respondents had expressed preference for languages that are emphasized in the curriculum and taught to them in the department, the researcher recommends that modern programming languages that can solve a variety of problems in addition to special-purpose programming languages be incorporated into Computer Science curriculums in our tertiary institutions.

## REFERENCES

[1] B. N. Miller, and D. N. Ranum, Problem Solving with Algorithms and Data Structures using Python. 2nd ed, Beedle & Associates, 2013.

[2] SL. Lee, What's the Big Deal about Coding? Figure 8: Nurturing for the future, Minimum theme on the framework, 2016.

[3] R. W. Sebesta, Concepts of programming languages. 10th ed, Pearson, New York, 2012.

[4] F. Hardaway, Why women shouldn't code, Stealthmode Corporations, 2015.

[5] K. Verspoor, The real reason more women don't code, The Conversation Africa Pilot, May 2016

[6] K. R. Parker, T. A. Ottaway, T. Joseph, J. T. Chao, J. Chang, "A formal language selection Process for introductory programming courses," Journal of Information Technology Education, Informing Science Institute, Santa Rosa, USA, vol. 5, pp. 133-151, 2006

[7] L. Mannila, and M. de Raadt, An objective comparison of languages for teaching introductory programming, in Proceedings 6th Baltic Sea Conference on Computing education research (Baltic Sea '06): Koli Calling 2006, ACM New York, NY, USA, pp. 32-37.

[8] S. Al-Imamy, J. A. Ajman, A. Mohamed, M. A. Nour, "On the development of a programming teaching tool: the effect of teaching by templates on the learning process," Journal of Information Technology Education Informing Science Institute, Santa Rosa, USA, vol. 5, pp 271-283, 2006.

[9] C. Edmondson, "Real women don't write programs," ACM SIFCSE Bulletin, ACM New York, NY, USA , vol. 40, iss. 2, pp 112-114, 2008.

[10] M. R. Wick, "Using programming to help students understand the value of diversity," in proceedings of the 40th ACM technical symposium on computer science education (SIGCSE '09), Chattanooga, TN, USA, ACM, New York, NY, pp. 367-371, March 2009.

[11] A. Yacob, and M. Y. M. Saman, "Assessing level of motivation in learning programming among engineering students," in Proceedings the International Conference on Informatics and Applications (ICIA2012), Malaysia, pp 425-432, 2012.

[12] U. Çakıroğlu, "Analyzing the effect of learning styles and study habits of distance learners on learning performances: A case of an introductory programming course," The International Review of Research in Open and Distributed Learning, vol. 5, iss. 4, Athabasca University, Canada, 2014.

[13] A. Asif, "Why learn programming?", The ICT Room, Pakistan.