Computer Science for Educators-Course Proposal

Fatima Bacha

Sheila Connelly

Kimberly Crowley

Johanna Guerrero

New Jersey City University

October 27, 2015

Global Educational Technology

Computer Science for Educators

CED201

**Course Level** 200 level

**Credits**-- 3 credit course

**Catalog Description**

**CED 201:** **Computer Science for Educators:** Introduces students to computer science as a discipline. This course presents an introduction to computers and programming. Topics include, but are not limited to, mathematical operations, searching techniques and algorithm analysis, procedures and functions, looping, Scratch, robotics, and web design.

**Course components**

The course will be an online course. If possible, there will be groups of students who will meet for the robotics unit.

**Rationale**

Since the country of Estonia made the decision to add computer coding to their national curriculum in 2012, other countries have made the decision to add coding to their national curriculums including Finland, England, and Australia. In the United States, Computer Science was officially recognized as being a STEM subject which will open the possibility for computer science to increase its presences in schools in the United States.

This global push for more computer science in schools will be bolstered by having teachers who are capable and comfortable with computer science. This course aims to give educators the knowledge and experience to fill the demand to teach courses in computer science and/or incorporate computer science in their regular course content.

**Student Learning Outcomes**

1.) Develop proficiency in the five technology competencies:

· Programming

· Hardware

· Data communications

· Data management

· Systems/software analysis and design

2.) Recognize and understand the professional, social, and ethical responsibilities associated with computer science.

3.) Apply tools, theory, and practices of Computer Science to a wide range of applications

4.) Develop intrapersonal skills for working effectively on teams to design and implement solutions to computational problems

5.) Develop interpretive skills:

· Analyze data statistically

· Interpret result of an experiment

· Draw reasonable conclusions based on experimental results

6.) Prepare for continued professional development. (University-wide)

7.) Develop effective written and oral communication skills. (University-wide)

**Instructional Procedures**

Students will have weekly readings and assignments. Students will respond to the assigned questions and respond to at least two classmates on the discussion forum on Blackboard Collaborate by Sunday at 11:59 pm. (Student will present in groups and individually) Participation, attendance. (Login to discussion board at least three times a week). There will be two major assessments. One assessment will be to create a lesson plan that would teach a computer science concept to students that does not require the use of a computer. The other assessment will be to work with a group to create a game using the programming language Scratch that you could use with your class. The game should use loops and decisions.

**Course Content**

**Week 1: Introduction to Computers:**

During week 1, educators will explore the impact of computers in our society, identify the need for teaching computer science in K-12. Educators will define a computer and its main hardware and software components.

**Week 2: Number Systems**

During week 2, educators will be introduced to number systems, decimal, binary, octal and hexadecimal. Educators will convert from base 10 to base 2 and vice versa and perform basic arithmetic operations such as binary addition and multiplication.

**Week 3: Introduction to programming**

In week 3, educators will acquire knowledge of various low level and high level computer programming languages used to communicate with computers .Educators will also be introduced to three computer programming languages: Visual Basic, C++ and Java.

**Week 4: Problem solving/Algorithms**

In week 4, educators will use problem solving to write algorithms. Educators will identify input, processing, and output and write the steps to carry out a task. Educators will utilize programming tools such as flowcharts and pseudocode to solve various problems.

**Week 5: Mathematical operations**

In week 5, educators will understand the concept of variable, data types and data assignment. Educators will be introduced to multiple operators and grouping symbols.

**Week 6: Mathematical Expressions**

During week 6, educators will be using basic mathematical operators and grouping symbols to build mathematical expressions

**Week 7: Making Decisions**

In week 7, Educators will utilize relational and logical operators to write *if/else statements*. Educators will write algorithms to solve various problems involving finding the greatest number, calculating averages and determining test scores.

**Week 8: Looping**

Educators will use increment and decrement operators in mathematical expressions and relational expressions. Educators will be introduced to three loop control structures: the *while* loop, the *do-while* loop, and the *for* loop

**Week 9: Procedures and functions**

During this week, educators will be introduced to modular programming by defining and calling functions and procedures. Educators will write a collections of statements to perform a task and gain understanding of local and global variables as well as passing data to functions and procedures.

**Week 10: Arrays**

In week 10, educators will use one Dimension and 2 dimensions arrays to store multiple values of same datatype. Educators will initialize arrays, accessing its elements, inputting and displaying array contents using a practical example related to storing grades and calculating a grade average.

**Week 11: Searching techniques and Algorithm analysis**

During week 11, educators will be introduced to search algorithms to locate a specific item in a collection of data. Educators will analyze linear search and binary searches algorithms then compare and contrast the two technique.

**Week 12: Web Design**

During this week, Educators will explore issues of social responsibility when using the internet as well as the impact and influence of the internet on today’s s society and on education. Educators will apply problem solving strategies to design their web page and will be using HTML and a variety of Styles of CSS to design their web pages based on their interests and experiences

**Week 13: Scratch**

During this week, educators will be introduced to the Scratch programming language, including the basic terms utilized in the language. Educators will practice the concept of event driven programming by writing Scratch stories and present them to the class. Peer reviews will be conducted.

**Week 14: Robotics**

During week 14, educators will be able to apply the programming and problem solving skills acquired from previous weeks. Educators will build and program robots. Students will be using either the LEGO Mindstorm NXT software to program Lego robots or using Scratch or C++ to program Finch robots to perform various tasks.

**Week 15: Graphing with Java**

Educators will utilize graphical tools to draw graphics in Java. Educators will write a java program using the Abstract Window Toolkit to draw a checkerboards and game boards and images onto a drawing panel.

**Undergraduate course**  
Educational Technology

**Assessment**

|  |  |
| --- | --- |
| Weekly readings and assignments | 40% of the course grade |
| Assessment 1: Lesson Plan | 20% of the course grade |
| Assessment 2: Group Scratch Game | 20% of the course grade |
| Participation | 20% of the course grade |

**Bibliography**

Bell, T., Witten, I., & Fellows, M. (n.d.). CS unplugged: An enrichment and extension programme for primary-aged students (2015 revision ed.) (S. Jarman, Ed.). Retrieved 30 October 2015, from <http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf>

CSTA K–12 Computer Science Standards. (2011). Retrieved 30 October 2015, from <http://www.csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf>

Gardiner, B. (2014, March 23). Adding coding to the curriculum. *The New York Times*. Retrieved from [HYPERLINK "http://www.nytimes.com/2014/03/24/world/europe/adding-coding-to-the-curriculum.html"http://www.nytimes.com/2014/03/24/world/europe/adding-coding-t HYPERLINK "http://www.nytimes.com/2014/03/24/world/europe/adding-coding-to-the-curriculum.html"o-the-curriculum.html](http://www.nytimes.com/2014/03/24/world/europe/adding-coding-to-the-curriculum.html)

Goode, J., & Chapman, G. (2015). Exploring Computer Science. Retrieved from <http://www.exploringcs.org/curriculum>

UK Department for Education (2013). Computing programmes of study: key stages 1 and 2. Retrieved 30 October 2015, from [https://www.gov.uk/government/uploads/syste HYPERLINK "https://www.gov.uk/government/uploads/system/uploads/attachment\_data/file/239033/PRIMARY\_national\_curriculum\_-\_Computing.pdf"m/uploads/attachment\_data/file/239033/PRIMARY\_national\_curriculum\_-\_Computing.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf)

Reges, S, & Stepp, M. (2007). *Building Java Programs* (2nd ed.). Pearson-Addison Wesley