School of Computing and Information Sciences

Course Title: Internship for Teaching Computational Thinking in K-12 **Date:** February 4, 2020

Course Number: CIS 1940

Number of Credits: 0

Subject Area: Programming	**Subject Area Coordinator:** Janki Bhimani **email:** jbhimani@fiu.edu
Catalog Description: Internship for teaching computational thinking to students in K-12, following established CS curriculum such as Google CS First, Code.org, and MIT Media Lab's Scratch.	
Textbook: Online Curriculum https://studio.code.org https://scratch.mit.edu https://csfirst.withgoogle.com/s/en/home	
References: Wing, J. 2006. Computational Thinking. CACM. 15, 5 (March 2006), 33-35	
Prerequisites Courses: None	
Corequisite Courses: None	

Type: Internship

Prerequisite Topics: None

Course Outcomes:

O1. Be able to follow plan lessons in computational thinking appropriate for the age-level and the prior knowledge of a K-12 student.
O2. Be able to teach how to plan a new program by breaking it down into smaller pieces, using storyboards, flowcharts, and pseudo code.
O3. Be able to teach how to use a programming tool such as MIT's Scratch or Code.org's app creation, to develop games and animations.
O4. Be able to provide feedback to K-12 students as they apply computational thinking skills.

This course should be overseen by FIU faculty that have experience in teaching computational thinking in programming classes.

School of Computing and Information Sciences
CIS 1940
Internship for Teaching Computational Thinking in K-12

Outline

Topic	Number of Lecture Hours	Outcome
• Follow plan lessons in computational thinking appropriate for the age-level and the prior knowledge of a K-12 student o Identify the various K-12 curriculum which exist o Select the most appropriate curriculum for the target group of K-12 student o Become familiar with the various lessons per level of student.	5	O1
• Teach how to plan a new program by breaking it down into smaller pieces o Explain what the purpose of a storyboard and how it is used to plan an animation. o Explain what pseudo code is and give basic examples of how to use it. o Explain how to create a flowchart, and explain how you would create it.	5	O2
• Teach how to use a programming tool targeted for K-12 children. o Select Google CS First, MIT Scratch, or Code.org ▪ Teach how to open, save, and test code ▪ Following curriculum, teach progressively harder programs.	15	O3
• Provide feedback to K-12 students as they learn computational thinking skills. o Provide positive reinforcement to each student o Help students identify their logic, syntax, and runtime errors. o Help students to "debug" their code o Help students correct errors in code o Motivate students to keep learning	10	O4

University students who register for this course will learn how to:

- Follow lesson plans for specific grade levels
- Teach how to breakdown a programming problem into smaller pieces
- Demonstrate how to design a program using pseudo code, storyboards, or flowcharts of the program's logic
- Explain how to write a program using CS tools targeted for children
- Monitor and give feedback to K-12 students as they practice computational thinking skills

School of Computing and Information Sciences
CIS 1940
Internship for Teaching Computational Thinking in K-12

Course Outcomes Achieved by

Internship in Teaching Computational Thinking in K-12

An internship in teaching computational thinking in K-12 will provide students with the following outcomes.

Outcome	
O1	Students will be able to follow lesson plans in computational thinking appropriate for the age-level and the prior knowledge of a K-12 student
O2	Students will learn how to teach breaking down a problem into smaller pieces
O3	Students will learn how to teach a programming tool targeted for K-12 children.
O4	Students will learn how to provide feedback to K-12 students as they learn computational thinking skills.

Oral and Written Communication:

- Written and oral discussions of how to teach computational thinking

Theoretical Contents:

- Abstraction
- Basic algorithmic thinking

Problem Analysis Experiences:

None

Solution Design Experiences:

- Weekly teaching internships, following lessons, programming with various CS tools