January 2017		March 2017		April 2017		May 2017		June 2017	
9	What is CS?	1	Problem Solving Walkthrough	3	Arduino Lab #2	8	String Practice	1	pygame intro
10	Introduction to CS	2	Practice with Problem Solving	4	Arduino Lab #2	9	Quiz On Strings Install Pygame	2	pygame intro
11	How Computer Made	3	Flowchart Quiz Installing Snap	5	Arduino Lab #2	10	Intro to Arrays	5	test
12	Binary System	6	Variables intro & flowchart->snap	6	Arduino Lab #2	11	Intro to Arrays	6	Final Project
13	Computer Memory	7	Practice Programs	7	test review	12	Intro to Arrays	7	Final Project
16	How code works	8	More Practice Programs	10	test (unit 1-4)	15	Arduino Lab #3	8	Final Project
17	Intro to Flowcharts	9	Comparison Operators	17	intro to python	16	Arduino Lab #3	9	Final Project
18	Practice Flowcharts	10	Logical Operators & Truth Tables	18	loops in python	17	Arduino Lab #3	12	Final Project
19	Structure Flowcharts	13	Introduction to If-Else	19	if, elif, else & methods	18	Arrays Quiz & Intro to Objects	13	Final Project
20	Practice Flowcharts	14	If vs if-else	20	drawing numbers practice	19	Intro to Objects	14	Final Project
		15	Practice Problems	21	star patterns	22	Hunt the Wumpus Example	15	Final Project
		16	Arduino Lab #1	24	star patterns	23	Hunt the Wumpus Example	16	Final Project
February 2017		17	Arduino Lab #1	25	variable scoping	24	Arduino Lab #4	19	Final Project
27	Holiday Assignment	20	Arduino Lab #1	26	python methods quiz	25	Arduino Lab #4	20	Final Project
28	Intro to Problem Solving	21	Introduction to Loops	27	hangman walkthrough	26	Arduino Lab #4	21	Final Project
		22	Code Analysis (factoring)	28	hangman walkthrough	29	Objects Quiz & pygame intro		
		23	Code Analysis (prime number)			30	pygame intro		
		24	Code Analysis (all primes to n)			31	field trip		
		27	Loop practice						
		28	Loop practice						
		29	Loop practice						
		30	Loop practice						
		31	Loop Quiz						

Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8

Unit 1: Algorithms

Unit Summary

This unit is meant to introduce the foundational concepts that is fundamental for the rest of the units in this course. First, the students will understand what is a computer and how it works. The students will also have an understanding of what computer science is and the scope it covers. Finally, the students will be introduced to the concept of algorithms and how to create and represent them (flowcharts).

Prescribed Learning Outcomes	Topics Covered	Assessments
The Nature Of ICT A8 A10 A11 **Introduction to Programming** B1 B2 B3 B4 B5 B6 B7 B8 **Programming Methodology** C1 C2 C3 C4 C5 C6 **Programming Methodology** D1 D2	• What is Computer Science? • How a computer works • What are algorithms? • How to represent algorithms • How to problem solve	**Diagnostic** • gallery walk on basic terminology **Formative** • in class problems **Summative** • computer parts quiz • flowchart quiz • flowchart of life assignment

Vocabulary: computer,computer science, computer engineering, hardware, software, programming, circuit, binary number, decimal number, bit, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, cache, RAM, hard drive, register, CPU, assembly, compiler, interpretor, flowchart, input, output, process, decision, sequence, selection, loop

Distribution of PLOs

	A8	A10	A11	B1	B2	B3	B4	B5	B6	B7	B8	C1	C2	C3	C4	C5	C6	D1	D2
1			X						X										
2		X	X						X										
3					X			X	X				X						
4			X	X															
5			X					X					X						X
6				X	X	X	X												
7												X	X			X			X
8												X	X			X			X
9												X	X			X			X
10												X	X		X	X			
11															X	X			
12	X											X	X			X		X	X
13	X									X	X	X	X	X	X	X	X	X	X
14	X									X	X	X	X	X	X	X	X	X	X
15	X									X	X	X	X	X	X	X	X	X	X

Day 1	Day 2	Day 3	Day 4	Day 5
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• identify the different fields of computer studies • how the different fields are related	• show how all fields are related in computer studies • clarify what is CS and it's subfields • give the focus of ICTP	• how computers are made from simple circuit to processor	• intro to binary • converting to and from binary • why computer uses binary system	• what is computer memory • how computer uses memory • the memory pyramid
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**
• identify, distinguish, and understand how the streams relate	• understand the objectives of this course • see the benefits of learning CS	• see the process of how a computer is built	• understand why computer uses binary • convert to and from binary	• understand the memory pyramid • understand how memory pyramid can affect computer speed
PLOs	**PLOs**	**PLOs**	**PLOs**	**PLOs**
A11 B6	A10 A11 B6	B2 B5 B6 C2	A11 B1	A11 B5 C2 D2
Language Components	**Language Components**	**Language Components**	**Language Components**	**Language Components**
• vocabulary ◦ computer,computer science, computer engineering, hardware, software, programming • mind map for the above words	• venn diagram to show how the vocabulary words from previous lesson connect	• vocabulary ◦ circuit	• vocabulary ◦ binary number, decimal number	• vocabulary ◦ bit, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, cache, RAM, hard drive, register
Overview	**Overview**	**Overview**	**Overview**	**Overview**
• welcome the class • vocabulary mind map and gallery walk ◦ cycle groups of 5	• review the mind maps from yesterday • create venn diagram to relate the different terms • go over subfields of CS	• show a series of videos demonstrating the different stages	• explain why computers use binary • converting to binary • converting from binary	• explain the different sizes of memory • explain the memory pyramid
Resources	**Resources**	**Resources**	**Resources**	**Resources**
• poster paper		• videos		

Day 6	Day 7	Day 8	Day 9	Day 10
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• How computer executes code	• what is a flowchart • the basic shapes and structures in a flowchart	• practice drawing and tracing flowcharts	• structured flowcharts • the 3 basic structures	• practice structured flowcharts
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**
• how computer understands code • difference between compiler and interpretor	• draw and follow a flowchart • understand the different shapes and structures	• be more comfortable with flowcharts	• understand why flowcharts need to be structured • recognize and structure unstructured flowcharts	• create structured flowcharts
PLOs	**PLOs**	**PLOs**	**PLOs**	**PLOs**
B1 B2 B3 B4	C1 C2 C5 D2	C1 C2 C5 D2	C1 C2 C5 D2	C1 C2 C4 C5
Language Components	**Language Components**	**Language Components**	**Language Components**	**Language Components**
• vocabulary ◦ CPU, assembly, compiler, interpretor	• vocabulary ◦ flowchart, input, output, process, decision		• vocabulary ◦ sequence, selection, loop	
Overview	**Overview**	**Overview**	**Overview**	**Overview**
• what the cpu does with the instructions • what is the difference between compiler and interpretor	• quiz on 1st 6 lessons • go through several examples of flowcharts • go through how to read flowcharts	• work period with practice problems	• go through examples of unstructured flowchart • go through how to convert them to structured flowchart	• work period with practice problems • holiday assignment
Resources	**Resources**	**Resources**	**Resources**	**Resources**
•		•		

Day 11	Day 12	Day 13	Day 14	Day 15
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• formative evaluation on holiday assignment • students given time to finish holiday assignment	• introduction to problem solving • solving on paper before trying on computer	• go through the steps of problem solving	• practice problem solving with the problem solving process	• install snap
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**
• get a formative assessment on holiday homework • complete holiday homework for tomorrow	• visualize their solutions on paper before seeing computer run simulation	• understand and apply the steps of problem solving	• apply the steps of problem solving and create solutions independently	• demonstrate understanding of flowcharts • create solutions independently
PLOs	**PLOs**	**PLOs**	**PLOs**	**PLOs**
C4 C5	A8 C1 C2 C5 D1 D2	A8 B7 B8 C1 C2 C3 C4 C5 C6 D1 D2	A8 B7 B8 C1 C2 C3 C4 C5 C6 D1 D2	A8 B7 B8 C1 C2 C3 C4 C5 C6 D1 D2
Language Components	**Language Components**	**Language Components**	**Language Components**	**Language Components**
	• vocabulary ◦ algorithm			
Overview	**Overview**	**Overview**	**Overview**	**Overview**
• work period for students to finish holiday homework	• students work in partners ◦ go through robo rally puzzles	• go over the steps of problem solving • go through some example problems with the steps applied	• work period for students to practice problem solving	• flowchart quiz • installing the program snap
Resources	**Resources**	**Resources**	**Resources**	**Resources**
•		•		

Unit 2: Variables, Data Types, and Basic Operators

Unit Summary

This unit introduces the students to the concept of variables and data types. It also covers the mathematical operators, comparison operators, and logical operators.

Prescribed Learning Outcomes	Topics Covered	Assessments

Programming Methodolgy
C1 C2 C3 C4 C5 C6 C7 C8 C9
 	
Programming Structures
D1 D2 D3
 	
Graphics and User Interfaces
E4
	• what are variables
	• the different data types
	  ◦ boolean, integer, float, string
	• mathematical operators
	  ◦ plus, minus, multiply, divide, modulus
	• comparison operators
	  ◦ <, >, ==
	• logical operators
	  ◦ not, and, or
	  ◦ truth tables
		Diagnostic
		• connect variable to math variables
		• recognition of operators
		
		Formative
		• in class problems
		
		Summative
		• quiz
		• test

Vocabulary: variable, data type, modulus, integer, float, string, boolean, less than, greater than, equal to, truth table, not, and, or

Distribution of PLOs

	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	E4
1	X	X	X	X	X	X	X	X	X	X	X	X	X
2-3	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X
5												X	

Day 1	Day 2-3	Day 4	Day 5
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• what are variables • why use variables • creating variables • data types • mathematical operators	• practice creating simple programs with variables and math operators	• introduction to comparison operators • comparison operators always return a boolean data type • students explore if-else block in snap	• introduction to logical operators • introduction to truth tables
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**
• understand how variables are stored in memory • understand the different data types	• create simple programs using variables and mathematical operators in snap	• apply comparison operators to their programs • discover how to use if else block in snap	• create truth tables • understand the 3 basic logical operators
PLOs	**PLOs**	**PLOs**	**PLOs**
C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 E4	C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 E4	C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 E4	D3
Language Components	**Language Components**	**Language Components**	**Language Components**
• vocabulary ◦ variable, data type, modulus, integer, float, string, boolean		• less than, greater than, equal to	• truth table, not, and, or
Overview	**Overview**	**Overview**	**Overview**
• explain what a variable is and how it works in memory • explain why we need variables • explain the different data types ◦ always stored as binary • go through math operators • give some example programs	• students practice making simple programs with flowcharts and snap	• introduce comparison operators • give some examples • students practice creating new programs	• go through the truth tables of the basic logical operators • give examples of connecting operators • students practice with truth tables
Resources	**Resources**	**Resources**	**Resources**
• snap	• snap	• snap	

Unit 3: If Statements

Unit Summary

This unit introduces the students on how to make the program do different things based on certain conditions. The students will learn if and if-else structures. They will also know how to nest the structures.

Prescribed Learning Outcomes	Topics Covered	Assessments
Nature of ICT A3 A4 A6 **Programming Methodolgy** C1 C2 C3 C4 C5 C6 C7 C8 C9 **Programming Structures** D1 D2 D3 D7 **Graphics and User Interfaces** E4	• if-else statement • if statement • creating programs with if and if-else statements • nested if and if-else statements	**Diagnostic** • recognize the if-else block in snap **Formative** • in class problems **Summative** • quiz • Arduino Lab • test

Vocabulary: if-else, if, ground, power, positive, negative, resistor, LED, breadboard, bus, circuit, short-circuit

Distribution of PLOs

	A3	A4	A6	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D7	E4
1				X	X	X	X	X	X	X	X	X	X	X	X	X	X
2				X	X	X	X	X	X	X	X	X	X	X	X	X	X
3				X	X	X	X	X	X	X	X	X	X	X	X	X	X
4-6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Day 1	Day 2	Day 3	Day 4-6
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• formal introduction to if-else statement • nesting of if-else statements	• introduce if statement • demonstrate difference between if and if-else	• Practice problems with if and if-else statements	• introduce students to circuits and arduino • writing simple program to control circuits
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**
• understand the if-else statement • nest if-else statements • able to create programs with if-else statements	• understand the difference between if and if-else • know when to use which	• create programs using if and if-else statements	• create simple circuits with arduino • create simple programs in snap to control the circuit • light up LED and push buttons
PLOs	**PLOs**	**PLOs**	**PLOs**
C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D7 E4	C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D7 E4	C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D7 E4	A3 A4 A6 C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D7 E4
Language Components	**Language Components**	**Language Components**	**Language Components**

• vocabulary
  ◦ if-else
	• vocabulary
	  ◦ if
			• vocabulary
			  ◦ ground, power, positive, negative, resistor, LED, breadboard, bus, circuit, short-circuit

Overview	**Overview**	**Overview**	**Overview**
• explain what if-else statement is • give example programs • practice problems	• explain what if statement is • compare and contrast if and if-else statements • give example problems • practice problems	• work period for practice problems	• students do lab with arduino
Resources	**Resources**	**Resources**	**Resources**
• snap	• snap	• snap	• arduino

Unit 4: Loops

Unit Summary

This unit shows students how the computer can repeat a segment of code. The students learn how to use a while loop. The students are shown how a to convert a for loop into a while loop and how to repeat a program forever.

Prescribed Learning Outcomes	Topics Covered	Assessments
Nature of ICT A3 A4 A6 **Programming Methodolgy** C1 C2 C3 C4 C5 C6 C7 C8 C9 **Programming Structures** D1 D2 D3 D7 **Graphics and User Interfaces** E2 E4 E5 E6	• what is a loop • when to use a loop • pattern recognition • nested loops • different type of loops (while, for, forever)	**Diagnostic** • recognize the forever block in snap **Formative** • in class problems **Summative** • quiz • Arduino Lab • test

Vocabulary: loop, iteration

Distribution of PLOs

	A3	A4	A6	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D7	E2	E4	E5	E6
1				X	X	X	X	X	X	X	X	X								
2				X	X	X	X	X	X	X	X	X								
3				X	X	X	X	X	X	X	X	X								
4				X	X	X	X	X	X	X	X	X								
5-8				X	X	X	X	X	X	X	X	X	X	X	X	X				
9				X	X	X	X	X	X	X	X	X	X	X	X					
10-13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Day 1	Day 2	Day 3	Day 4	Day 5-8	Day 9	Day 10-13
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• introduce a loop • explain how a loop works in flowchart and in snap • give example problems • tracing programs	• code walk through if finding all factors of a number	• code walk through of checking if integer is a prime number	• code walk through of finding all prime numbers from 1 to n • an example of nested loop	• practice problems with loops	• assess student understanding with quiz	• students learn how to use a seven segment LED • create a binary convertor
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**
• understand what a loop is • create a simple loop	• give a solution on how to find factors	• give a solution on how to verify prime number	• find all prime numbers from 1 to n	• create programs that use loops	• demonstrate knowledge and application of loops	• create a circuit that performs binary conversion
PLOs	**PLOs**	**PLOs**	**PLOs**	**PLOs**	**PLOs**	**PLOs**
C1-9	C1-9	C1-9	C1-9	C1-9 D1-3 D7	C1-9 D1-3	A3 A4 A6 C1-9 D1-3 E2 E4-6
Language Components	**Language Components**	**Language Components**	**Language Components**	**Language Components**	**Language Components**	**Language Components**
• vocabulary ◦ loop, iteration			•	•		
Overview	**Overview**	**Overview**	**Overview**	**Overview**	**Overview**	**Overview**
• explain how a loop works • explain how the different loop controls in snap • give example problems and trace examples	• have students come up with a solution • compare different solutions from the students	• have students come up with a solution • compare different solutions from the students	• have students come up with a solution • compare different solutions from the students	• students have work period to practice	• students have whole block to write quiz	• students do lab with arduino
Resources	**Resources**	**Resources**	**Resources**	**Resources**	**Resources**	**Resources**
• snap	• snap	• snap	• snap	• snap		• arduino

Unit 5: Methods

Unit Summary

In this unit, students transfer from snap to python. They also learn how to implement methods which were touched upon throughout the course as students naturally started modularizing their flowcharts.

Prescribed Learning Outcomes	Topics Covered	Assessments
Nature of ICT A3 A4 A6 **Programming Methodolgy** C1 C2 C3 C4 C5 C6 C7 C8 C9 **Programming Structures** D1 D2 D3 D4 D7 **Graphics and User Interfaces** E2 E4 E5 E6	• what is a method • what are parameters • methods returning values • program flow with methods	**Diagnostic** • recognize modularizing of flowchart **Formative** • in class problems **Summative** • quiz • test

Vocabulary: method, parameter, return, global variable, local variable

Distribution of PLOs

Day 1	Day 2	Day 3	Day 4	Day 5-6	Day 7	Day 8
Lesson Objectives	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**	**Lesson Objectives**
• introduction to python • variables, operators, and if-else statements in python	• how to do loops in python	• if-elif-else shortcut in python • introduce methods • parameters • returning data • tracing methods	• practice making methods • practice abstracting to make more efficient methods	• practice with methods via star patterns	• variable scope	• assess student understanding with quiz
Students are able to...	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**	**Students are able to...**
• convert their simple snap programs to python programs • convert flowchart to python	• convert their loop programs in snap to python • convert their flowcharts to python	• understand how nested if-else statements can be converted to if-elif-else • create methods • modularize	• abstract a pattern from drawing numbers ◦ make connection with lab #2	• abstract star patterns and reuse methods	• understand variables with same name in different methods are different variables	• demonstrate knowledge and application of methods
PLOs	**PLOs**	**PLOs**	**PLOs**	**PLOs**	**PLOs**	**PLOs**
C1-9 D1-D3	C1-9 D1-D3	C1-9 D1-D3	C1-9 D1-D3	C1-9 D1-3 D7	C1-9 D1-3	A3 A4 A6 C1-9 D1-3 E2 E4-6
Language Components	**Language Components**	**Language Components**	**Language Components**	**Language Components**	**Language Components**	**Language Components**
		• vocabulary ◦ parameter, method, return	•		• vocabulary ◦ global var, local var	
Overview	**Overview**	**Overview**	**Overview**	**Overview**	**Overview**	**Overview**
• demonstrate how to convert flowchart to python • demonstrate how to convert snap program to python	• demonstrate how to convert a flowchart and snap programs that contain loops to python	• demonstrate how to convert nested if-else to if-elif-else • creating and using methods • parameters	• students have a work period to figure out how to draw numbers on a command line and create an abstract way	• work period for students to practice using methods	• give several examples of variable scoping ◦ have students figure out	• students have whole period to write the quiz
Resources	**Resources**	**Resources**	**Resources**	**Resources**	**Resources**	**Resources**
• snap	• snap	• snap	• snap	• flowgorithm	• flowgorithm	

Unit 6: Strings

Unit Summary

In this unit, students learn how to concatenate strings, get substrings, and get specific characters in a string.

Prescribed Learning Outcomes	Topics Covered	Assessments

	• what is a string
	• how to concatenate strings
	• how to get parts of a string
	  ◦ substring
	  ◦ single characters
	• how to find how many characters are in a string
		Diagnostic
		• recognize a string
		
		Formative
		• in class problems
		
		Summative
		• quiz
		• test

Vocabulary:

Distribution of PLOs
