Computer Science Lesson Plan

Hello! Please consider my computer science unit plan. This unit plan utilizes Google Classroom, Edhesive software, compiler, laptops, and internet (see unit plan at the bottom of this document). The community resources include Google Classroom because this is where students can interact with each other, and many other platforms allow scholars to primarily interact with the teacher. All of the resources students are using are digital. The Edhesive curriculum provides a platform of videos, problems, formative and summative assessments, etc. for students to review at home and in class. The compiler this unit will be Code Skulptor, an online compiler that is accessible even at home, and can be used to compile data that can be further checked in the Edhesive software for confirmation that students have solved the given problems. There are other people that they will have access to, particularly in the videos (Ada Lovelace teaches some content through the videos), including retired professional basketball player and computer science major Chris Bosh. These will provide content and inspiration for students to continue their educational pursuits and keep them interested.

CONTEXT: I have 4 SPED students, with an additional student I am looking to see if a SPED label will be issued, 7 I have personally labeled as GATE (unofficial), and 0 ELL labeled students. The school has about 550 students, 120 of them being freshmen, and is primarily of Latino ethnicity, majority female, and a charter school under LAUSD. Please see below for specific and general approaches to each class.

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 1: 8.1 What are Functions? And 8.2 Creating Functions**
CS Standard #	**3A-AP-18**
CS Standard	**Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.**

Lesson Objective	Students will be able to: create their own subprograms or functions
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration to prove that they understand the material is the way I will keep them moving. This is done by giving them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing

	them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.
Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. My evaluative questions will be incorporated into the time students are getting guided practice time. During that time, I will ask questions that help me see if students understand the material and where scaffolding or differentiation may be needed. I will also use independent practice time to check for understanding with certain questions posed to them and me evaluating their answers so they can move on. At that time, students will be asked to explain steps. The exit ticket will reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, Laptops, internet**	5 mins	"Please submit your Do Now answers within 5 minutes, which is by 8:05am." 1) What does the while loop in the code below do? (--> represents indentation) inFile = open ("words.txt", "r") line = inFile.readline() while line: --> print (line) --> line = inFile.readline() inFile.close() / Door to Do Now strategy	Students will complete the Do Now section

Class Discussion / **Google Classroom, Laptops, internet**	10 mins	1. Create an original python code(the language they are learning) and use it to randomize the scholars' names to see what the partners will be 2. Say, "Follow the partners list that were randomly put together. Look at your partner's answer in Google Classroom. You now have one minute to critique your neighbor and convince them to change their answer to match yours. Leave notes. Go!" 3. Once the minute is done, show them the "Do Now example" picture for 30 seconds. "This is just an example of what you should know how to do, using arrows to get the right answer." 4. Then ask, "I'm going to choose someone to show us how they got their answer. You will need to use arrows to show us the progression of your answer." (4-6 mins) / Routine, technology use for peer to peer learning instead of turn and talk	Each student shares with their partner based on the instructions
Intro to 8.1 What are Functions? / **Edhesive software, compiler, laptops, internet**	10min	"Raise your hand if you have ever heard me use the term "function"? What is a function?" "Today, we will learn another word for "function" called "subprogram." Everyone say 'subprogram.'" "Take notes on these bullet points for 90 seconds and then study the vocabulary words posted on the board if you are finished" / Ask questions and use discovered learning techniques Be energetic and introduce interesting stories of functions doing incredible things	Properly define a function as a set of commands (given a name) Students are taking notes and contributing to and leading class discussions
CODE EXPLORATION / **Edhesive software, compiler,**	10 mins	1) What is the module? 2) What functions do you recognize? 3) What does float mean?	Students answer the prompts and then solve the problem using code

laptops, internet		4) Do you have to have a prompt in this code for it to work? import math a = float(input ("Number:")) print (math.sqrt(a)) / Provide prompts	
Lesson Practice (3 Q's) / **Edhesive software, compiler, laptops, internet**	5 mins	"Complete the lesson practice. You have 4 minutes. If you finish, start the code practice on the next page. Start now." /	Students complete the lesson practice questions
CODE Practice / **Edhesive software, compiler, laptops, internet**	10 mins (HW)	Write a program that asks the user for a number. If the number is between 1 and 255, the program outputs the corresponding ASCII character. If it is outside of that range, the program outputs "Out of range."	Students use functions, such as chr(), print(), and input() to create working code Students use their own creativity to create a program that meets the requirements, including developing their own variable names, input prompts, etc.
Intro to 8.2 Creating Functions / **Edhesive software, compiler, laptops, internet**	10min	"Does anyone remember seeing def before?" / Encourage student and peer to peer conversation	Students join in class discussion, as well as peer to peer conversation to talk through what type of functions they want to create
CODE EXPLORATION / **Edhesive software, compiler, laptops, internet**	10 mins	1) What method does def define below? 2) Why does "Code in the SubProgram" print after "Code in Main"? def example (): print ("Code in the SubProgram") # ******* MAIN ******* print ("Code in Main")	Students answer the prompts and then solve the problem using code

		example() / Provide prompts	
Lesson Practice (6 Q’s) / **Edhesive software, compiler, laptops, internet**	5 mins	“Complete the lesson practice. You have 5 minutes. If you finish, start the code practice on the next page. Start now.” / Use formative assessment to gage understanding	Students complete the lesson practices (formative assessment)
CODE Practice / **Edhesive software, compiler, laptops, internet**	10 mins (HW)	“Write a function named ilovepython that prints out I love Python three times. Then, call that function.” / Demonstrate how to use the concepts learned to complete a problem like this	Students use functions, such as chr(), print(), and input() to create working code Students use their own creativity to create a program that meets the requirements, including developing their own variable names, input prompts, etc.
EXIT SLIP / **Google Classroom, compiler, laptops, internet**	5 mins	Provide EXIT TICKET: 1) Print the rhyme below using subprograms: This little pig went to the market. This little pig stayed home. This little pig had roast beef. This little pig had none. This little pig cried wee wee wee wee! All the way home.	Students choose a subprogram, such as print(), to print the words

Computer Science Lesson Plan

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 2: 8.3 Parameters / 8.4 Chris Bosh on Functions**
CS Standard #	**3A-DA-10**
CS Standard	**10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.** **12 - Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.**

Lesson Objective	Students will be able to: Successfully show an understanding of Unit 7 material through an assessment
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration that they understand the material is the way I will keep them moving. This is done by getting them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.
Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. My

	evaluative questions will be incorporated into the time students are getting guided practice time. During that time, I will ask questions that help me see if students understand the material and where scaffolding or differentiation may be needed. I will also use independent practice time to check for understanding with certain questions posed to them and me evaluating their answers so they can move on. At that time, students will be asked to explain steps. The exit ticket will reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, compiler, laptops, internet**	5 mins	"Please complete your Do Now questions by 8:05am before we exchange with our neighbor." / Door to Do Now	Students will complete the Do Now section online
Class Discussion / **Edhesive software, compiler, laptops, internet**	15 mins	1. Create and use python code to randomize the scholars' names to see what the partners will be 2. Say, "Follow the partners list that were randomly put together. Look at your partner's answer in Google Classroom. You now have one minute to critique your neighbor and convince them to change their answer to match yours. Leave notes. Go!" 3. Once the minute is done, show them the "Do Now example" picture for 30 seconds. "This is just an example of what you should know how to do, using arrows to get the right answer." 4. Then ask, "I'm going to choose someone to show us how they got their answer. You will need to use arrows to show us the progression of your answer." (4-6 mins) / Routine, technology use for peer to peer learning instead of turn and talk	Each student shares with their partner based on the instructions

Computer Science Lesson Plan

Intro to 8.3 Parameters / Edhesive software, compiler, laptops, internet	20min	“What do we call information sent to a function...what are two synonyms for “function”?... You have used parameters before” “Everyone say ‘Parameter.’” “Take notes on these bullet points for 90 seconds and then read the slide when you finish”	Students are taking notes and contributing to and leading class discussions
CODE EXPLORATION / Edhesive software, compiler, laptops, internet	20 mins	“Answer the questions on the board. You have 4 minutes.” Questions to answer: 1) What is the module? 2) What functions do you recognize? 3) What does float mean? 4) Do you have to have a prompt in this code for it to work? import math a = float(input ("Number:")) print (math.sqrt(a)) / Provide prompts for students to answer questions and think about the problem	Students answer the prompts and then solve the problem using code
8.3 Lesson Practice / Edhesive software, compiler, laptops, internet	5 mins	“Complete the lesson practice. You have 4 minutes. If you finish, start the code practice on the next page. Start now.”	Students complete the lesson practices (formative assessment)
8.3 CODE Practice / Edhesive software, compiler, laptops, internet	10 mins (HW)	“Read the instructions on the board silently to yourself. You have 90 seconds.”	Use of functions, such as chr(), print(), and input() Students use their own creativity to create a program that meets the requirements, including developing their own variable names, input prompts, etc.
EXIT SLIP / Google Classroom, compiler,	5 mins	“Please turn in your EXIT TICKET within the next five minutes. The person in the fourth row will put up the	Exit tickets are submitted and students show that functions are a type of

laptops, internet		laptops for their column when students are done and everyone will remain at whisper level and in their seats until I dismiss you. Start your Exit ticket now.” 1) How do you define a "subprogram" in your own words?	subprogram

Computer Science Lesson Plan

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 3: 8.4 Chris Bosh On Functions / Unit 8 Quiz / 8.5 Functions – Returning Values**
CS Standard #	**3A-AP-18**
CS Standard	**Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.**

Lesson Objective	Students will be able to: use optional parameters and the return function as they create their own subprograms
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration that they understand the material is the way I will keep them moving. This is done by getting them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student. Providing a virtual word wall in the beginning of class can help reinforce high frequency terms they have learned as well.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.
Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. My

	evaluative questions will be incorporated into the time students are getting guided practice time. During that time, I will ask questions that help me see if students understand the material and where scaffolding or differentiation may be needed. I will also use independent practice time to check for understanding with certain questions posed to them and me evaluating their answers so they can move on. At that time, students will be asked to explain steps. The exit ticket will reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, compiler, laptops, internet**	5 mins	Say, "Please submit your Do Now answers within 5 minutes, which is by 8:05am." 1) Write a program below that has an optional parameter / Door to Do Now strategy	Students complete the Do Now The while loop prints a line and then reads the next line and repeats until the lines are all read in the document named "words.txt"
Class Discussion / **Edhesive software, compiler, laptops, internet**	10 mins	1. Create an original python code (the language they are learning) and use it to randomize the scholars' names to see what the partners will be 2. Say, "Follow the partners list that were randomly put together. Look at your partner's answer in Google Classroom. You now have one minute to critique your neighbor and convince them to change their answer to match yours. Leave notes. Go!" 3. Once the minute is done, show them the "Do Now example" picture for 30 seconds. "This is just an example of what you should know how to do, using arrows to get the right answer." 4. Then ask, "I'm going to choose someone to show us how they got their answer. You will need to use arrows to show us the progression of your answer." (4-6 mins)	1. Each student has shared with their partner. 2. Answers have been written down by students and they have discussed an accurate answer

Computer Science Lesson Plan

		/ Routine, technology use for peer to peer learning instead of turn and talk	
New Procedure + Quiz 8 / Edhesive software, compiler, laptops, internet	30min	"After the quiz, I want you to take your quiz here, staple it and leave it at the table faced down like this. Once you do that, return to your seat, take out your notebook and write the terms and definitions that are on the board in your notebook. (Student) will demonstrate for us." / Provide instructions and demonstrations for what to do	Students complete the quiz and follow the procedure properly. Participate in the demonstration of what to do
8.5 Functions – Returning Values / Edhesive software, compiler, laptops, internet	25 mins	"You will need to know this. The variable inserted becomes the parameter for this function. In the subprogram where I define that function, the parameter is replaced with the one in the main program. Because there are three parameters in the main and sub programs, the first parameter in the main program replaces the first one in the subprogram. This is the same for the two corresponding parameters. If you do not enter any of the last two parameters, the default will be used. Let's practice." Use arrows to trace the code, evaluating the output of the code.	Students show their understanding of the replacing of variables and parameters in the main program and subprogram.
Lesson Practice / Edhesive software, compiler, laptops, internet	5 mins	"Complete the lesson practice. You have 4 minutes. If you finish, begin the exit ticket on the next page. Start now." / Provide formative assessment for data collection	Students complete the lesson practices (formative assessment)
EXIT SLIP / Google Classroom, compiler, laptops, internet	10 mins	"In order to do the exit ticket, let's go over this problem. The variable inserted becomes the parameter for this function. In the subprogram where I define that function, the parameter is replaced with the one in the main program. This parameter becomes the variable inside the	Students perform proper calculations based on the problem. Students ask for help and get support from each other.

		program and is not changed based upon the code. This is similar to the problem in your Exit Ticket. Raise your hand if you need help.”	

Computer Science Lesson Plan

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 4: 8.6 Example – Using Several Functions / 8.7 Tracing Code**
CS Standard #	**3A-AP-18**
CS Standard	**Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.**

Lesson Objective	Students will be able to: create our own subprograms or functions
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration that they understand the material is the way I will keep them moving. This is done by getting them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.
Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. My evaluative questions will be incorporated into the time students are getting guided practice time. During that time, I will ask

	questions that help me see if students understand the material and where scaffolding or differentiation may be needed. I will also use independent practice time to check for understanding with certain questions posed to them and me evaluating their answers so they can move on. At that time, students will be asked to explain steps. The exit ticket will reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, compiler, laptops, internet**	5 mins	"Please submit your Do Now answers within 5 minutes, which is by 8:05am."	Students read the code, produce the correct answer
Class Discussion / **Edhesive software, compiler, laptops, internet**	10 mins	1. Create an original python code (the language they are learning) and use it to randomize the scholars' names to see what the partners will be 2. Say, "Follow the partners list that were randomly put together. Look at your partner's answer in Google Classroom. You now have one minute to critique your neighbor and convince them to change their answer to match yours. Leave notes. Go!" 3. Once the minute is done, show them the "Do Now example" picture for 30 seconds. "This is just an example of what you should know how to do, using arrows to get the right answer." 4. Then ask, "I'm going to choose someone to show us how they got their answer. You will need to use arrows to show us the progression of your answer." (4-6 mins) / Routine, technology use for peer to peer learning instead of turn and talk	Correct other's work with feedback, understand functions and terms. Lead correspondence with partners
8.6 Example – Using Several Functions /	35min	Say, "Underline the second variable. Circle the second parameter. Draw a line from the variable to the	Show understanding of the use of several subprograms

Edhesive software, compiler, laptops, internet		parameter.” / Provide this prompt and use it to check for understanding	Track how an input value impacts the code as it is manipulated as it is used by multiple functions in the code Identify local variables
8.7 Tracing Code / Edhesive software, compiler, laptops, internet	10 mins	Say, “Show me with a thumbs up or thumbs down if this is the error in the code” / Ask for scholar feedback for steps in tracing code	Trace the code to find the error, considering local variables as well
CODE Practice / Edhesive software, compiler, laptops, internet	10 mins (HW)	Provide a prompt for scholars to write down and use to complete their assignments	Take notes on the prompt Complete the code practice for homework Students use their own creativity to create a program that meets the requirements, including developing their own variable names, input prompts, etc.
EXIT SLIP / Google Classroom, compiler, laptops, internet	5 mins	Provide exit tickets assessments and assess data 1. How do you recognize a local variable? 2. What is one use of a local variable?	Demonstrate understanding of the use of local variables by completing the exit ticket and put up laptops

Computer Science Lesson Plan

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 5: Unit 8 Test Prep**
CS Standard #	**3A-DA-10**
CS Standard	**10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.** **12 - Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.**

Lesson Objective	Students will be able to: Successfully show an understanding of Unit 8 material through an assessment
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration that they understand the material is the way I will keep them moving. This is done by getting them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.

Computer Science Lesson Plan

Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. My evaluative questions will be incorporated into the time students are getting guided practice time. During that time, I will ask questions that help me see if students understand the material and where scaffolding or differentiation may be needed. I will also use independent practice time to check for understanding with certain questions posed to them and me evaluating their answers so they can move on. At that time, students will be asked to explain steps. The exit ticket will reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, compiler, laptops, internet**	5 mins	Greet students at the door and ask them to get a laptop and complete the Do Now. When the bell rings, say, "Please complete your Do Now questions on Google Classroom in 5 minutes." When 5 minutes are done, say "Turn in your Do Now." Wait 30 seconds for them to do that. 1) What is output if the user enters -3? (Use the attached picture.) **Answer**: 20	Students have shown understanding of how to trace code
Class Discussion / **Edhesive software, compiler, laptops, internet**	10 mins	1. Create an original python code (the language they are learning) and use it to randomize the scholars' names to see what the partners will be 2. Say, "Follow the partners list that were randomly put together. Look at your partner's answer in Google Classroom. You now have one minute to critique your neighbor and convince them to change their answer to match yours. Leave notes. Go!" 3. Once the minute is done, show them the "Do Now example" picture for 30 seconds. "This is just an example of what you should know how to do, using arrows to get the right answer." 4. Then ask, "I'm going to choose someone to show us how they got their answer. You will need to	Each student has completed the Do Now and understand the reasoning of the correct answer

		use arrows to show us the progression of your answer.” (4-6 mins) / Routine, technology use for peer to peer learning instead of turn and talk	
Grade Unit 8 Quiz / Edhesive software, compiler, laptops, internet	15 mins	1. (Pass out quizzes by placing them faced down) “Flip the quiz over and put your name on the bottom of the quiz using ‘CB: name’. Place an ‘X’ if the answer is incorrect, check mark if correct. At the end, write the total number of correct answers on the top of the front of the quiz and circle it.” 2. Write the answers on the board as seen to the right. 3. After they correct the quiz, tell them, “write the total number of correct answers on the top of the front of the quiz and circle it.” 4. “Pass the quizzes to the end of the row and I will get them from there.” 5. Erase the answers.	Correctly mark quizzes and writing the total correct answers on the front paper and circling it and passing them down the rows. Quiz 8 Solutions: 1. a 6. c 2. d 7. d 3. b 8. c 4. d 9. a 5. c 10. a
8.3 and 8.5 Lesson Practices / Edhesive software, compiler, laptops, internet	10mins	Say, “Log into Edhesive and complete the 8.3 Lesson Practice. You have 4 minutes” “Complete the 8.5 Lesson Practice. You have 4 minutes”	Students complete the lesson practices (formative assessment) Students are contributing to and leading class discussions
Unit 8 Test Prep (Vocabulary: 20 mins; Test prep doc: 30mins) / Edhesive software, compiler, laptops, internet	50 mins	“You will use the next 3 minutes to review your vocabulary words. Everyone, go to the Unit 8 Vocabulary Practice in Edhesive. If you get there before I walk around and confirm everyone is there, you should take those moments to begin your review. Begin.” “Open the Vocabulary Story document on Google Classroom. Use each vocabulary word to create a story. You have 8 mins. Go!” “Open the test prep document. Begin working on that	Vocabulary is put into sentences effectively Students work on the test prep document until 5 minutes before class ends

		document until 5 minutes before the end of class. Start now." / Facilitate learning through challenging scholars to use the terms to create an original story.	
EXIT SLIP / **Google Classroom, compiler, laptops, internet**	5 mins	"Please complete your EXIT TICKET on Google Classroom within the next five minutes. When you finish, place your laptop in the cart and plug them up. Start now." 1) What is output if the user enters -6?	Students have shown understanding of how to trace code through completing the exit ticket

Computer Science Lesson Plan

Teacher Name	**Mr. Tolan**
Course Name (Grave Level)	**Intro to Computer Science (9)**
Date of Lesson	**DAY 6: Unit 8 Test**
CS Standard #	**3A-AP-18**
CS Standard	**Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.**

Lesson Objective	Students will be able to: Successfully show an understanding of Unit 8 material through an assessment
Differentiation	Keeping GATE students busy will be important, so letting them work on code following a demonstration that they understand the material is the way I will keep them moving. This is done by getting them instructions to go ahead of me and I'll catch up to them. The Zone of Proximal Development calls for some more challenging problems to build their mastery beyond what they already know. They will be able to read the problem, begin coding and engage in productive struggle before I help them. Introducing them to application problems and scenarios can bring out some higher level DOK understanding, which can possibly translate to their future projects, and will be done with the Exit Ticket. Some students will be given the opportunity to have extra time to perform tasks. The lesson involves various ELD proficiency levels because students get to practice speaking to one another using academic language based on prompts. Students can speak to other students who speak the same language and even use translators via the computer. Google translate may be necessary for some students who need a better understanding of English. Some videos can help break the communication barrier that may exist, while other students may be willing to help if they sit next to the student. Students get to see my modeling and another student to learn from, who may most likely be an advanced student.
Theories Applied	Theories applied include Vygotsky and Bruner's theories concerning Zone of Proximal Development (ZPD) and scaffolding, Skinner's theories concerning conditioning, Piaget's theories concerning pursuits of intelligence, etc.
Assessment	My lesson will begin with Do Now practice problems to assess their knowledge of prior lessons and presented content focuses. The class will mainly consist of a test, so that formative assessment will provide data that can be assessed afterward. The exit ticket will

	reveal if they successfully understood the material presented in class and apply it to solve real world problems.

Lesson Activity (in order)/Resources	**Time Duration**	**Directions to students/ Pedagogy**	**Student Action**
DO NOW / **Google Classroom, compiler, laptops, internet**	5 mins	"Please submit your Do Now answers within 5 minutes, which is by 8:05am." 1. What is output if the user enters 2?	Students have shown understanding of how to trace code
Class Discussion / **Edhesive software, compiler, laptops, internet**	5 mins	"Can someone show us how they got their answer? You will need to use arrows to show us the progression of your answer." Ask for thumbs up or down if they agree Have students discuss for further understanding / Student led activity that allows students to teach each other and solidify understanding in the one teaching	1. Each student has completed the Do Now and understand the reasoning of the correct answer 2. One student does the work for the students to demonstrate how they arrived at the answer 3. Other students critique the answer of the student through approval and discussion
Test Prep / **Edhesive software, compiler, laptops, internet**	15mins	Say, "Open your test prep Document you completed for homework. Please compare your answers to mine. For number 1, what did you bring? I brought this answer. Number 2 what..."	Students are giving answers to problems they have looked at for homework
Test Unit 8 / **Edhesive software, compiler, laptops, internet**	50min	Say, "After the test, please silently perform the tasks posted on the board in order." Tasks on picture: 1. Start the 9.1 What Are Arrays notes using Google Classroom (Use headphones to listen to the video) 2. Submit any other work not yet submitted (Projects, homework, etc.)	Students complete the test and follow the procedure properly.
EXIT SLIP /	10 mins	"Please complete your EXIT TICKET on Google	Students perform proper calculations

Google Classroom, compiler, laptops, internet		Classroom within the next five minutes. When you finish, place your laptop in the cart and plug them up. Start now." 1) What is output if the user enters 4?	based on the problem.

Further Explanation of Classroom Approaches

Learning by Design is "a design challenge and attempting a solution using only prior knowledge -- individually and/or in small groups" (Kolodner, 1998). The students will use turn and talk methods in the beginning of the lesson during the Guided Practice. This will help students generate ideas of executing the problem at hand. They will also be partnered and given time to fill in the blank with terms from a word bank and asked to get the desired result. They will do this using the compiler software, the code they will be given as a prompt, and a document that scholars will fill out on their computers and turn in electronically. They will be asked to achieve certain output from some predetermined pieces of code, as well as code they develop themselves. The outputs will be mathematical in nature. In some cases, there will also be terminology that students must match and identify with definitions. This helps to see the parallels between coding and math. They will have access to videos, in some cases homemade, for their own review. This allows students to further show and develop their learning in applying the information they have learned in the videos at home to solve a problem. The videos are important because they introduce the scholars to coding so that they can be aware of it once they come to the next class. The lesson is meant to full of "cycles of exploratory and experimental work, followed by reflection on what has been learned, application of what was learned to achieving the design challenge, evaluation of that application, and generation of additional learning issues" (Kolodner, 1998). The reflection happens when students record the steps they took to find the correct answers and then as we talk together as a class. Learning by Design shows that there should be "authentic, engaging design-and-build activities that enliven students' interest," and in this case, in computer science (Kolodner, 1998). This is meant to occur during the fill in the blank partner activity, but mainly when they must use code to show that they can achieve the correct answers to the problems. I will also be able to put a group's program on display for parent teacher conferences, administration, display on the walls of the room, etc. Learning will be assessed through the use of the compiler that checks to see if scholars have developed a program that meets the criteria that was given to them. They will also be assessed through the group talks concerning their fill in the blank assignments.

"Future work into situated cognition, from which educational practices will benefit, must, among other things, try to frame a convincing account of the relationship between explicit knowledge and implicit understanding" (Brown, 1989, 15). In other words, the idea behind situated cognition is that there should be a context developed that create a greater depth of learning. In doing so, this approach becomes relevant to the pursuits of cognitive apprenticeship. "Cognitive apprenticeship attempts to develop densely textured concepts out of and through continuing authentic activity" (Brown, 1989, 13). This is relevant when you want the real life theme to promote greater learning. The real life application in our lesson plans include the use of code to solve problems, for example, that relate to finding letters or words in text or having to have a capitalized letter in a password. This is done in the code practices. "Goal-based scenarios allow students to pursue well-defined goals and encourage the learning of both skills and cases in service of achieving those goals" (Schank, 1992).The side effect is that scholars can end up learning unrelated details in order to learn the intended material. This can provide some context for the development of scholars' knowledge, and have them build out a broader knowledge base. Making sure I do not use media without concern for how I may overload the students with it can lead for media to have a negative association or impact with my students (Mayer, 2003).

In coding, a simulation context can be helpful. A problem is presented as if the scholars have to solve a math problem in a real life scenario. They would be presented coding that is meant to be used to solve a problem so that, for example, a company can present a minimum viable product (MVP) to venture capitalists for investments. This could mean they must trace code, with guidance from video support, to diagnose the problem and then create steps to prevent future problems. The code would not be something they need to be familiar with, but introduce them to coding principles and even some of the terms used in coding that are used in math, but with a different meaning in a different context. This could pair the learning in a goal-based scenario and use situated cognition for enhanced learning. In doing so, students also create knowledge through a learning by design methodology.

Using computers in my class is a great way to do coding. When it comes to differentiated instruction, students have the ability to get help based on their level of progression. One way is the use of resources available for learning syntax of the language. This can come in the form of accessing documents that explain the use of certain syntax. "Computer-assisted learning, in which educational software helps students develop particular skills, is particularly promising, especially in math" (Horn, 2019). This would be provided either through the access of guides at the websites that dedicated to the language that I am teaching or another resource. I must also consider that having "small progressively sequenced tasks" can help students' attention become activated to complete the tasks at hand (Danley, nd, 14; Merrill, 2002, 46). The computer will let my students get evaluation by various students through a written form and interactive platform. This can change the dynamic because I can assign partners randomly and vicinity is not a limitation. I can also see progress of students on the computer through my own in real time in order to provide differentiated instruction through data I can obtain immediately. I can also collect and compile data immediately and make adjustments in the class that same day and period. This can improve my ability to impact the dynamics of the class. I can also assign varying assignments to meet the differences in needs, which helps me build my capacity (Noguera, 2017). It is helpful that I also need to make sure I have a coherent system for delivering instruction (Noguera, 2017). This will be performed through the use of computer software that allows students to track and review the number of questions they have gotten right or wrong on assessments, see immediate feedback on code practice assignments and various other resources. The use of a blendedand flipped model lets students have watch videos of the lesson at home already and prepare them to talk about the content on a deeper level for further understanding and application in class (Knewton; The University of Texas). The class also incorporates software applications, networks, computers, and internet in daily practices and management (Edutopia, 2007). These imply the use and access of materials from home outside of class for their convenience and reference. This also helps bridge an equity gap that allows for students to study at a pace they can help control even more.

Using class periods of about one hour and thirty minutes, the objective of the unit is for scholars to review some common functions they have already used and to understand how they simply code writing. It will incorporate a two-week unit intended to reach this goal:

- Week 1:
    - DAY 1-2: 8.1 What are Functions? And 8.2 Creating Functions
    - DAY 3-4: 8.3 Parameters / 8.4 Chris Bosh on Functions
    - DAY 5: 8.4 Chris Bosh On Functions / Unit 8 Quiz / 8.5 Functions – Returning Values
- Week 2:
    - DAY 6-7: 8.6 Example – Using Several Functions / 8.7 Tracing Code
    - DAY 8-9: Unit 8 Test Prep
    - DAY 10: Unit 8 Test

Sources:

- Brown, J.S., Collins, A., & Duguid, P. (1989). *Situated Cognition and the Culture of Learning*. Center for the Study of Learning, University of Illinois at Urbana-Champaign.
- Danley, B., James, N., Mims, C., &Simma, A. (n.d.). *Behaviorism Theory: And Its Relation to Instructional Design*. Mercer University. http://faculty.mercer.edu/codone_s/tco363/2014/behaviorism.pdf
- Horn, Michael. (2019). *New Research Answers Whether Technology is Good or Bad for Learning*. Aurora Institute. https://aurora-institute.org/cw_post/new-research-answers-whether-technology-is-good-or-bad-for-learning/
- In *What Is Successful Technology Integration?: Well-integrated use of technology resources by thoroughly trained teachers makes twenty-first-century learning possible.* (2007, November 5). Edutopia. https://www.edutopia.org/technology-integration-guide-description
- Knewton. *The Flipped ClassroomInfographic*. (n.d.). https://alliant.instructure.com/courses/13627/files/669345/download?wrap=1
- Kolodner, J., Crismond, D., Gray, J., Holbrook, J. &Puntambekar, S. (1998). *Learning by Design from Theory to Practice*. EduTech Institute and College of Computing, Georgia Institute of Technology. http://www.cc.gatech.edu/projects/lbd/htmlpubs/lbdtheorytoprac.html
- Mayer, R. E., & Moreno, R. (2003). *Nine Ways to Reduce Cognitive Load in Multimedia Learning*. Educational Psychologist, 38(1), 43-52.
- Merrill, M. D. (2002). *First principles of instruction*. Educational Technology Research and Development, 50(3), 43-59.
- Noguera, Pedro. (2017, April 19).*Pedro Noguera, Ph.D. - Education for a Changing Society (04/17/17)*. Family Action Network. https://www.youtube.com/watch?v=0q2Jc9dgjjA&feature=youtu.be
- Schank, R. (1992).*Goal-Based Scenarios*. The Institute for the Learning Sciences. http://cogprints.org/624/1/V11ANSEK.html
- Smith, Megan. (2016). *Computer Science For All*. Accessed: 7/12/20. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all
- The University of Texas at Austin. *Flipped Classroom*. https://facultyinnovate.utexas.edu/flipped-classroom