LATHROP ENGINEERING

Name:

UNIT 3: CONTROL STRUCTURES

AP Computer Science A Unit Due Date: **October 18, 2019**

Welcome to the third unit of *AP Computer Science*! In this unit, we'll expand on the basics of Java programs as we explore the different control structures used in programming: *if, if-else, else-if, for,* and *while*. Each of these control structures will give us a different tool for handling special circumstances in our programming problem solving. In the end, the expectation is that you learn the following:

- The differences between *if, if-else,* and *else-if* control structures and how to use each
- The differences between *while* and *for* loops and how to use each
- How to combine control structures for more complex program solutions
- When different types of control structures should be used

As we move through this unit, you are responsible for making adequate progress through the assignments, and for being done by the Unit Due Date (**October 18, 2019**). You are also responsible for completing each part before moving on to the next. Our unit is broken up into three main parts:

Part 1: **Decision Making** *(20 pts) Approx. 3 days*		
The start of this unit looks even more closely at the use of the *if, if-else,* and *else-if* control structures. We started using them back with Jeroo, and continued with some simple uses in the last unit. Here you'll get a chance to use each in various kinds of programs. In this first part to the unit, we'll also make sure that we really understand the differences between these structures.		If/If-Else/Else-If Notes
		Complete 4 Decision Tasks
		Choose Your Own Adventure!
		Check-off from Mr. Benshoof

Part 2: **Loops** *(50 pts) Approx. 3 days*		
Loops let us create chunks of program that repeat themselves. We can use *while* loops to repeat until a condition is met, or we can use *for* loops to repeat a set number of times. Each of these control structures is essential for making programs that are more robust and useful than simple decision machines.		For/While Notes
		Complete 8 Looping Tasks
		Take the Unit 3 Quiz
		Control Structures Assignment
		Check-off from Mr. Benshoof

Part 3: **Gradebook** *(30 pts) Approx. 2 days*		
In the last part of this unit, you'll be asked to make a small gradebook program that can help you calculate your own grades. This program will need to let you enter a variety of assignment scores, store them, and then let you know how many A's, B's, C's, etc you earned, as well as your overall average. More complex programs like this will require a mix of loops and conditionals!		Gradebook Brainstorm Notes
		Create Your Own Gradebook
		Write Log 3: *Control Structures*
		Check-off from Mr. Benshoof

(20 pts) Approx. 2 days

This unit is all about controlling the *flow* of a program. Old programming languages from the early days of computers ran purely *sequentially* – from one line of code, on to the next, then the next, etc. Eventually programmers developed ways to jump around within their code, creating a mess of programming lines often referred to as "spaghetti code". The most useful tools for controlling the order in which the computer thinks through problems ended up being *if*, *if-else*, *else-if*, *while*, and *for* structures. The first part of this unit is about the **conditionals** that let us make choices between what to do in the program.

1. Start by watching the three introductory videos on our website. These videos provide another simple overview of different kinds of if statements and how they can work with primitives, strings, or multiple Boolean questions.
2. Take a full page of good notes on the different details discussed in the videos. Make sure that your notes include information about how if, if-else, and else-if differ from one-another!
3. Now, complete the next 4 Java Tasks below that all require making decisions!
   a. JAVA TASK 12: Write a program that lets the user indicate whether they have bread, peanut butter, jelly, milk, and a knife in their cupboard. Based on the results, your program should tell the user whether or not they have what they need for a minimally acceptable PB&J sandwich. The logic in your program should be based on one ***if-else*** statement.
   b. JAVA TASK 13: Use a series of ***if*** statements to make a program that lets the user enter a positive integer less than 20, and then tell the user all of the prime factors of that number. Remember that % can help you see what numbers evenly divide each other.
   c. JAVA TASK 14: Use a series of ***else-if*** statements to make your own daily reminder. The user should be able to enter the day of the week as a *String*. Your program should then have an output that tells them when they need to get up in the morning, what the school schedule is like for that day (if any), and one special thing happening that day.
   d. JAVA TASK 15: Go back and add a fourth equation to your calculator program from Unit 2. This new equation should have multiple variables in it, and should allow the user to solve for any of those variables. This will require a few extra ***if*** statements.
4. **Choose your own adventure!** Now, create a (simple) 3-step "choose your own adventure" story. The story should be complete up to three reader-decisions. If each decision has 2 options, that means there will be 8 different story paths you'll need to create.

Part 1: Tasks	*5 points*	*4-3 points*	*2-1-0 points*
Conditionals Notes	+ Watch all presentations + You took a full page of notes on conditionals in Java + Your notes include detailed differences between the three conditional formats	- Less than a full page of conditionals notes - No notes on differences in conditionals	- Very brief or no notes in your notebook
	10 points	*9-4 points*	*3-0 points*
Java Tasks 12-15	+ You completed all 4 Java Tasks from this section	- You did not complete all 4 tasks	- You did not complete any tasks
	5 points	*4-3 points*	*2-1-0 points*
Choose Your Own Adventure!	+ You created a 3-step Choose Your Own Advtenture!	- Your adventure does not have 3 full steps	- You did not create a choose your own adventure.

(50 pts) Approx. 3 days

We've gotten a lot of experience with asking questions of different kinds and using various conditionals to make decisions in a program. The second half of control structures is all about looping within a program. Here, we'll get good practice with both the *while* and *for* loops, also called "sentinel controlled" and "counter controlled" repetition, respectively.

1. Watch the three videos on looping control structure. Take a full page of good notes on each, making special note of the differences between the different kinds of looping structures, how, and when they are used!
2. Now, complete the following Java Tasks below that all require using Scanners and/or if statements!
   a. JAVA TASK 16: Write a program that lets the user enter as many positive integers as they would like, and responds by telling them each time whether the numbers is even or odd. Your program should use a *while* loop, and should let the user type a negative number to end the program.
   b. JAVA TASK 17: Write a program that uses a *for* loop to ask the user for 7 integers. The program should then print both the sum and the decimal average of the seven numbers.
   c. JAVA TASK 18: Write a program that lets the user enter as many positive integers as they'd like and counts how many are even and how many are odd. The user should be able to type a negative number to end the program.
   d. JAVA TASK 19: Write a program that uses a ***for*** loop to print a grid of numbers that show integer values multiplied by 10, 100, 1000, and 10000.

      Example Output:

n	n*10	n*100	n*1000	n*10000
1	10	100	1000	10000
2	20	200	2000	20000

   e. JAVA TASK 20: Write a program that lets the user enter a maximum integer. The program should then print a nice 4x8 grid of *random* numbers that are all less-than-or-equal-to the user defined max number.
   f. JAVA TASK 21: Write a program that prints a grid of numbers that display a multiplication table up to 12x12. This program should use a nested ***for*** loop to make the grid and calculate the proper values.
   g. JAVA TASK 22: Write a program that lets the user enter a positive single digit integer. The program should then use a nested for loop to print a grid of alternating 0's and 1's. The length of the side of the grid should be the value the user typed. The 0's and 1's should alternate in a checkerboard pattern.

      Example Program: User: 3

      (Output)

1	0	1
0	1	0
1	0	1

   h. JAVA TASK 23: Write a program that lets the user enter two small integers (less than 15), and then prints a rectangle of asterisks (*) that has a width and height equal to the user's two numbers.
3. Complete the *Control Structures Assignment* about the different control structures we need to know
4. Take the Unit 3 Quiz **by October 11** – the quiz is linked from our website!

Part 2: Tasks	*10-7 points*	*6-4 points*	*3-0 points*
Conditionals Notes	+ Watch all presentations + You took a full page of notes on loops in Java + Your notes include detailed differences between while & for	- Less than a full page of loops notes - No notes on differences in types of loops	- Very brief or no notes in your notebook
	20-15 points	*14-5 points*	*4-0 points*
Java Tasks 16-23	+ You completed all 8 Java Tasks from this section	- You did not complete all 8 tasks	- You did not complete any tasks
	10-7 points	*6-4 points*	*3-0 points*
Control Structures Assignment	+ You completed the Control Structures Assignment	- You only did part of the Control Structures Assignment	- You did not do the assignment
Unit 3 Quiz: Control Structures	+ You took the Unit 3 Quiz by the due date + Your grade is based on how many you get correct.	N/A	- You did not take the Unit 3 Quiz

(30 pts) Approx. 2 days

The final part of our unit asks you to create a program to meet the needs of you and your fellow students. Very often, when programmers are asked to create a program for someone else, they are given the parameters (criteria) and the programmer gets to decide *how* to make the program work. Here, you've been tasked with creating a small gradebook application that can help students summarize and check their grades.

1. Brainstorm different ideas for how your Gradebook program can work. Write a full page of notes, ideas, and thoughts about how to organize your program to meet the needs of the client. The program you are being asked to make must:
    a. Let the user enter a variety of assignment types (at least 2: *quizzes* and *homework*).
    b. Let the user enter a variety of assignment grades
    c. Let the user enter the grades and assignments in any order
    d. When the user is done entering assignments, the program should tell them some information:
        i. How many assignments were listed for each type (how many quizzes, how many homeworks, etc)
        ii. How many of the total list of assignments were A's, B's, C's, D's, or F's
        iii. What the average score (based on either points OR percentages… whichever you like best) on all assignments was
        iv. Assuming the upcoming final would be worth 15% of the course grade (and all entered assignments were the remainder of the grade), what grade would the student need on the final to earn their desired course grade.
    e. The program should be relatively easy to use and easy to understand.

2. *Log 3: Control Structures* The last part of our unit should be a 1-page response to the ideas of Java and control structures. How do you define the 5 control structures? How do you see them as different from each other? What possible uses can you think of for each? Look up what a *switch* is and compare it to our control structures. Look up a *do-while* and compare it to our control structures. How is class going so far? What questions do you still have that you haven't gotten answered yet?

Part 3: Tasks	*5 points*	*4-3 points*	*2-1-0 points*
Gradebook Brainstorm Notes	+ You wrote a full page of brainstorms, ideas, and notes about how to structure your program + Your notes outline a ***PLAN*** for your program	- You wrote less than a page - Your notes do not outline a coherent plan	- Your notes are lacking or missing - There is no plan for your program
	15-10 points	*9-5 points*	*4-0 points*
Create Your Gradebook	+ You created a gradebook program that meets all the criteria listed above. + Your gradebook works as intended	- Your gradebook program does not meet some criteria - Your gradebook program mostly works	- Your gradebook program does not work at all
	10-8 points	*7-4 points*	*3-0 points*
Log 3: *Control Structures*	+ You wrote a complete page in your engineering notebook + Your response addresses all the questions in the prompt provided	- You wrote less than a full page	- You wrote less than half a page
Checkoff from Benshoof	+ Mr. Benshoof got to see your calculator work successfully!	N/A	N/A