















UNIT 4: OBJECT ORIENTED DESIGN

Welcome to the fourth unit of *AP Computer Science*! In this unit, we'll look at class structure and how classes, objects, and methods interact to make Object Oriented Design possible. In addition, we'll look more closely at different kinds of methods and how to make methods that will do the things we want them to! In the end, the expectation is that you learn the following:

- How to make your own classes, methods, and objects
- How to use constructors in the creation of your own classes and objects
- The "Is-A" and "Has-A" relationships
- How to define methods such as constructors, getters, setters, and mutators

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 (**November 8, 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: Classes & Constructors (30 pts) Approx. 3 days	
The start of this unit is all about making some new classes and helping define them with constructors. A simple class can do a lot for program organization, and a good constructor can help you define specific variables along the way! Here in this first part of the unit, we'll look at how Object Oriented Design (OOD) is used to organize programs and help create the logic that will drive our larger software design!	 OOD Notes
	 Constructor Notes
	 6 Constructor Tasks
	 6 Coding Bat Challenges
	 Check-off from Mr. Benshoof
Part 2: Getters, Setters & Mutators (50 pts) Approx. 3 days	
In the second part of the unit, we'll learn about the main kinds of methods that classes use: <i>getters</i> , <i>setters</i> , and <i>mutators</i> . Each of these kinds of methods serve specific functions, and by combining them in a smart way we can make our programs do cool things really easily. Here, we'll also look in more detail at the <i>method signature</i> and learn what all the different parts mean so that we can start to understand other people's programs better too!	 Getter/Setter/Mutator Notes
	 Complete 8 Methods Tasks
	 6 Coding Bat Challenges
	 Take the Unit 4 Quiz!
	 Check-off from Mr. Benshoof
Part 3: LHS Zoo (30 pts) Approx. 2 days	
The final part of our unit asks you to create an accounting program for the new Lathrop Zoo! All zoos need to keep track of their animals and supplies, and here you've been asked to create a program that the new zoo keepers can use to manage their zoo resources. You'll need to plan out the class structure, including which objects should be represented with classes, and what actions can be represented with methods!	 LHS Zoo Brainstorm Notes
	 Create Your LHS Zoo Program
	 Write Log 4: OOD
	 Check-off from Mr. Benshoof

(30 pts) Approx. 3 days

This unit covers the basics of *object oriented design*, and in particular the role that classes, constructors, and methods play in the development of a larger program. So far, every program we’ve written has been contained within a single class. In reality, programs are often built with many classes that all talk and interact with each other. This allows us to consider the different parts of our program as ‘objects’ within the program, and tell them to do different things that we’ll define as special ‘methods’.

The first part of this unit has us make our first classes and constructors. **Classes** are the structures that let us define types of things (like Jeroos), and their corresponding **constructors** give us a way to define important variables upon making those objects (like new Jeroo(2,3,SOUTH,35)).

1. Start by watching the three introductory videos on classes, variables, and constructors. Watch closely, because these concepts will be the foundation of the rest of the year!
2. Take a full page of good notes on classes, variables, and constructors. Make sure that your notes include details on the class signatures (how to define classes), private vs. public variables (when to use each), and constructor signatures (how to define constructors).
3. Now, complete the next 6 Java Tasks below that are all about defining classes, their variables, and their constructors!
 - a. JAVA TASK 24: Write a class called ASSIGNMENT that could be used to keep track of single assignments in a gradebook program. Include at least 5 variables and 1 constructor.
 - b. JAVA TASK 25: Write a class called CAR that could be used to keep track of individual cars in a database. Include at least 4 variables of 2 different types, and 1 constructor.
 - c. JAVA TASK 26: Write a class called SHAPE that could be used to keep track of individual shapes for a geometry program. Your class should include at least 3 variables including 3 different variable types, and 1 constructor.
 - d. JAVA TASK 27: Write a class called SPORT that could be used to keep track of individual sports at a big event like the Olympics. Your class should have 4 boolean variables and 1 constructor.
 - e. JAVA TASK 28: Write a class called SCHOOL that the school district could use to keep track of information about their schools here in Fairbanks. Your class should have 10 variables that include some ints, some doubles, some Strings, and some Booleans. You should also have 1 constructor.
 - f. JAVA TASK 29: Write a class called ACCOUNT that a bank could use to keep track of user accounts. Your class should have at least 4 variables and 1 constructor.
4. Lastly, create an account using your student s# e-mail at the website codingbat.com. This website gives a cool way to start working with single methods, and will even test your methods for you! Go to the Java programming site (not the Python site), and go to Warm-up 1. Pick any 6 challenges and complete them (all test cases green!).

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 classes and constructors + Your notes include detailed information on how to make constructors and class variables	- Less than a full page of notes on classes and constructors - Your notes are missing important parts	- Very brief or no notes in your notebook
 Java Tasks 24-29	+ You completed all 6 Java Tasks from this section	- You did not complete all 6 tasks	- You did not complete any tasks
 Coding Bat	+ You created an account at codingbat.com + You completed 6 WarmUp#1 challenges in codingbat.com	- You only completed 4 or 5 codingbat.com challenges	- you completed fewer than 4 codingbat challenges
 Checkoff from Benshoof	+ Mr. Benshoof got to see your Java programs run successfully	N/A	N/A



(50 pts) Approx. 3 days

Now that we’ve learned how to make the basic framework for a class – signature, variables, constructors – it’s time to start adding methods. Methods in a class come in three main varieties:





Getters are used to ask for and return the value of class variables

Setters are used to set or change the values of specific class variables

Mutators are used to make any other things happen within or between different objects

In this part of the unit, you’ll learn to write a variety of methods and then use a corresponding **Driver (or Runner)** class to confirm that your methods are working.

1. Watch the three videos on *Getters & Setters*, *Static Methods*, and *More About Methods*. Take a full page of good notes on these topics. Make sure that your notes include details about the specific method signatures used by these different methods, as well as the difference in intent behind each one.
2. Now, complete the following Java Tasks below that all require creating your own classes and methods! For every class below, write the class indicated *and include at least 3 variables, at least 1 constructor, all setters and getters, as well as at least 3 mutators*. The mutators in these classes **do not** have to be functional: their signatures should be appropriate, but the body of the method can simply be a comment describing the intended interaction or purpose of the method.
 - a. JAVA TASK 30: Write the class REFRIGERATOR following the above guidelines that could keep track of the contents of your refrigerator at home.
 - b. JAVA TASK 31: Write the class CAT following the above guidelines that could keep track of your pet cat’s day.
 - c. JAVA TASK 32: Write the class BUS following the above guidelines that could be used to keep track of school busses for the school district.
 - d. JAVA TASK 33: Write the class APCOURSE following the above guidelines that could be used to keep track of AP Courses and their parameters here at Lathrop.
 - e. JAVA TASK 34: Write the class VEGETABLE following the above guidelines that could be used by a grocery store to keep track of vegetable-related information.
 - f. JAVA TASK 35: Write the class TEACHER following the above guidelines that could be used by a student-made program to keep track of their teachers during the school year.
 - g. JAVA TASK 36: Write the class STUDENT following the above guidelines that could be used by a teacher-made program to keep track of their students during the school year.
3. Lastly, go back to CodingBat.com and log in with the e-mail you provided when you made the account. Go to the Java resources and the “String 1” section. Pick any 6 challenges from the String 1 group and complete them!




Part 2: Tasks	10-7 points	6-4 points	3-0 points
 Getter/Setter/Mutator Notes	+ Watch all presentations + You took a full page of notes on different kinds of methods in Java + Your notes include details about getters, setters, and mutators	- Less than a full page of methods notes - No notes on method signatures	- Very brief or no notes in your notebook
 Java Tasks 30-37	+ You completed all 8 Java Tasks from this section	- You did not complete all 8 tasks	- You did not complete any tasks
 Coding Bat	+ You completed 6 String #1 challenges in codingbat.com	- You only completed 4 or 5 codingbat.com challenges	- you completed fewer than 4 codingbat challenges
 Take the Unit 4 Quiz!	+ You took the Unit 4 Quiz before the due date + Your grade is based on the number correct	N/A	- You did not take the Unit 4 Quiz by the due date



(30 pts) Approx. 2 days

The last part of our unit will ask you to create a variety of classes that could interact with each other in simple ways. You’ve been asked to create a cataloging program that will help the Lathrop administration keep track of animals, supplies, and employees in our brand new Lathrop Zoo! To do this, you’ll need to make a program that fits the following criteria:

1. Start – as usual – by brainstorming different ideas for your Lathrop Zoo. What kinds of animals, supplies, or employee types are there to make classes for? What meaningful variables might they have?
2. Create your program such that:
 - a. You have at least 5 different animal classes (representing different kinds of animals).
 - b. Each animal class should have at least 4 private class variables of at least 2 different types
Pro Tip: pick variables that might be related to the supplies you’ll want to use
 - c. Your animal should have a method called “makeNoise()” that causes it to print the proper sound
 - d. Your animal classes all have a toString() method.
 - e. Your animal should have at least 2 other methods (with any name) that logically change the class variables for that animal.
 - f. Your animal classes all have the proper setters/getters
 - g. You have at least 4 different zoo supply classes (representing different kinds of supplies).
 - h. Each supply class should have at least 3 private class variables of at least 3 different types
 - i. Your supply classes have all the proper setters/getters
 - j. Your supply classes all have a toString() method.
 - k. You have at least 2 different employee classes (representing different kinds of zoo employees)
 - l. Each employee class has at least 3 private class variables of at least 3 different types
 - m. Your employee classes all have the proper setters/getters
 - n. Your employee classes each have 2 methods that take a zoo animal object AND a zoo supply object and use them to change proper instance variables. For example, you might have one employee take an animal and some food in a method called feedTheAnimal(Animal a, Food f). In this method, the animal’s hunger variable might go down, as well as the quantity variable of the food.
 - o. Make a Driver (or Runner) class to test your different methods and make sure your classes, objects, methods, and variables are all working like you expected!
3. *Log 4: Object Oriented* The last part of our unit is to write a full page response about object oriented design.

Part 3: Tasks	5 points	4-3 points	2-1-0 points
 LHS Zoo Brainstorm Notes	+ You wrote a full page of brainstorm, ideas, and notes about how to structure 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 LHS Zoo Program	+ You created the LHS Zoo program that meets all the criteria listed above. + Your LHS Zoo Program works as intended	- Your LHS Zoo program does not meet some criteria - Your LHS Zoo program mostly works	- Your LHS Zoo program does not work at all
	10-8 points	7-4 points	3-0 points
 Log 4: <i>Object Oriented Design</i>	+ You wrote a complete page in your engineering notebook	- You wrote less than a full page	- You wrote less than half a page

