

CSC 130

Programming Assignment #1

Due Date: by 11:55pm Saturday, September 27th

Quiz: Monday, September 29th

Objectives:

- Design and implement multiple classes using inheritance
- Practice manipulating an array of objects
- Practice developing program logic

Overview:

You will be developing a program to simulate **traffic on a one-lane road** (implemented using an array) containing two types of vehicles: **Cars** and **Buses**. Collisions can occur between Cars and Buses, between Cars and Cars, and between Buses and Buses.

Problem Decomposition:

Both Cars and Buses will have an associated horsepower, Cars will have a color, and Buses will have a weight. The result of *collisions* between vehicles is outlined below:

1. **Car vs. Bus**
 - If a car tries to move into a bus's space, the **car must stop** (bus stays).
 - If a bus moves into a car's space, the **bus pushes the car out** (car is removed).
2. **Car vs. Car**
 - If a car tries to move into another car's space:
 - If they are **different colors**, they "crash" → both are removed, and a new car is placed at a random location on the road.
 - If they are the **same color**, the car with greater horsepower survives (the other car is removed).
3. **Bus vs. Bus**
 - If a bus tries to move into another bus's space:
 - The **heavier bus** survives (the other bus is removed).
 - If equal weight, both remain in place.

Coding and Testing Your Solution:

Once you've settled on a design, you will want to think about the behaviors (i.e. methods) each class will need. You will then need to think about how to test your solution. Data to the program will consist of values between -1 and 1 (-1, 0, 1) with each resulting in the following behavior:

- -1: the vehicle at the current location will move back (from index 3 to 2 for example)
- 0: the vehicle at the current location will not move (stays at the current location)
- 1: the vehicle at the current location will move forward (from index 4 to 5 for example)

You are encouraged to create an input file that will test all possible scenarios including edge cases. I will create an inputfile for grading, and all submissions will be executed using the same data file.

Plagiarism Policy:

This assignment is to be completed **individually**. It is intended to help you understand the concepts of creating and manipulating an array of objects, use inheritance and to build your problem-solving skills. You can receive help from another person, but the kind of help that is acceptable should be verbal. You can explain a concept to someone else and use drawings to assist in this explanation, but you should never look at someone else's code, nor should you allow another student to look at yours. Additionally, GenAI tools can be used, but you must indicate what tool you used and how you used it. For example, maybe you used ChatGPT and asked it a series of questions. You will need to include the name of the tool and the prompts/questions posed in a text file submitted to Brightspace. Please note that these tools respond best with specific/detailed prompts. Additionally, you will want to make sure that you understand the code provided by the tool.

Style/Coding/Documentation Requirements:

- You should spend some time making your output as neat as possible. The overall appearance of the output will be graded.
- Neatness Counts!!! Proper indentation is required. Be consistent with your use of white space.
- Be sure to select meaningful variable names.
- There should be a Javadoc comment at the top that includes a title, description and your name. Javadoc comments should precede all methods and must be formatted according to Javadoc standards. Include line comments as needed to explain what your code does. You do not need to comment every line of code.

Submission:

Your submission will be in three parts: 1) your code; 2) the diagram of your class hierarchy along with method definitions, and 3) a link to a video of you explaining your project. See details below:

Code (20%)

You will upload all the code and your dataset to Brightspace for grading. I must be **able to download your code, run it and have the code work as expected**. Output should be displayed after each collision. Your output must be sufficient for me to determine which vehicles have collided and that the result of the collision is as expected.

Problem Decomposition (20%)

You will upload a .jpg or .png image summarizing your problem decomposition to Brightspace. Specifically, how did you take your large problem and break it into classes/methods that helped to solve the larger problem. Your image should clearly identify the class name, the instance variables as well as the methods defined (including a brief description, what data is passed to the method, and what if anything is returned).

Explanation Video (40%)

You will upload a link to a 5-minute video of you explaining the project on Brightspace. Your video should include the following:

Video quality/Details:

- Record the video with both your face and your presentation/code.

- Provide either a link to the video in a **google drive folder** (that is public to anyone with the link) or to a private **Youtube** video.

The 5-minute video should contain:

- (1 min) A walk through of the code that is executed when a Car collides with a Bus
- (1 min) A walk through of the code that is executed when a Car collides with a Car
- (1 min) A walk through of the code that is executed when a Bus collides with a Bus
- (1 min) A discussion of how inheritance was used in this assignment
- (1 min) A brief description of the values you chose for testing your code (including edge cases) and why

Submitting Files:

You will be submitting 3 things to Brightspace

1. A zip file containing your source files (all .java files) and your data file
2. A PNG or JPEG image of your project decomposition – if you can't see your image render in Brightspace I won't be able to see it either
3. A text file that contains the link to your video and a summary of the resources you used (for example, URLs to websites, the name of the GenAI tool used (including the prompts), etc.)

Quiz (20%)

An in-class quiz will be administered to assess your understanding of the project. You will not be allowed to look at your code during the quiz.