

**CS102 – Algorithms and Programming II**  
**Programming Assignment 3**  
**Spring 2023**

**ATTENTION:**

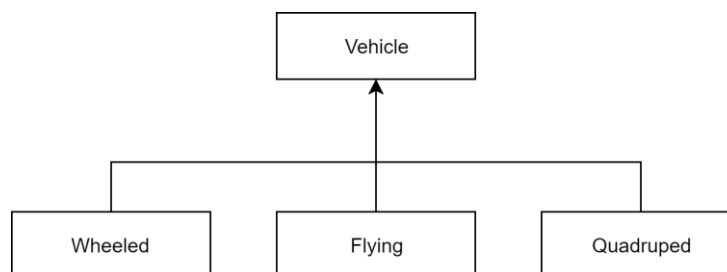
- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:  
**CS102\_Sec1\_Asgn3\_YourSurname\_YourName.zip**
- Replace the variables “Sec1”, “YourSurname” and “YourName” with your actual section, surname and name.
- You may ask questions on Moodle and during the scheduled Zoom meeting.
- Upload the above zip file to Moodle by the deadline (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TAs and tutors during the scheduled Zoom meeting.

**GRADING WARNING:**

- Please read the grading criteria provided on Moodle. The work must be done individually. Code sharing is strictly forbidden. We are using sophisticated tools to check the code similarities. The Honor Code specifies what you can and cannot do. Breaking the rules will result in disciplinary action.

**Race Simulation**

In this assignment, you are going to implement a Java program to simulate a turn-based race of autonomous vehicles. Different types of vehicles are categorized based on the following diagram:



Each vehicle has a name (string), speed (float), current position (float) and fuel (int). Different types of vehicles move with different factors of their speed on various road types, vehicle's speed is multiplied by a factor based on the road type. There are three types of roads: asphalt, dirt, and stone. The following table includes the speed factors for different road and vehicle combinations:

Road \ Vehicle	Wheeled	Flying	Quadruped
Asphalt	1.00	1.00	0.50
Dirt	0.75	1.00	1.00
Stone	0.75	1.00	0.75

For example, when going on dirt, a wheeled vehicle moves with  $speed * 0.75$ . Since this is a turn-based simulation, the vehicle's speed at that turn is determined based on where the vehicle is at the beginning of the turn. Do not change the speed if the vehicle enters a different type of road during that turn. For example, suppose a wheeled vehicle is on asphalt, and should move 30 units, but after 10 units stone road begins. In this case, the vehicle will not change speed, and move the remaining units on the stone road.

In vehicle class, implement a `void move(int roadType)` method that moves the vehicle based on the current type of road (0: asphalt, 1:dirt, 2: stone). This method should first check the current fuel of the vehicle, if it is positive, then move the vehicle accordingly. In case fuel is not positive, this method does nothing. If this method moves the vehicle, decrease its fuel by one.

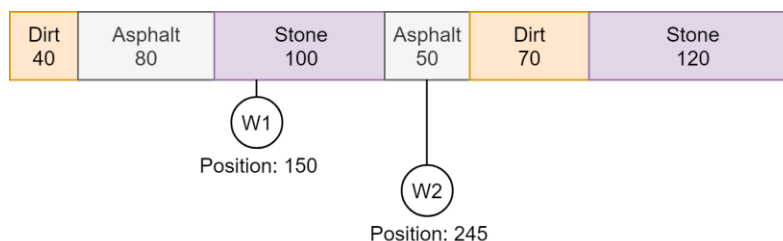
Although we include an implementation for the move method in the vehicle class, we will override the move method in the subclasses that extends vehicle. To this end, override the move method in wheeled, flying and quadruped classes based on the behavior of the corresponding vehicles on the given road types. For example, the move method in wheeled class should check the given road type, and if it is 2 (stone), multiply the speed by 0.75 and add it to the position. In short, given a vehicle of any type and the road, you need to:

- Check if fuel is positive, do nothing if it is not.
- Move the vehicle by  $speed * factor$ , where factor is found based on given road type.
- Decrease fuel by one.

In your simulation class, keep an int array for roadType and a float array for roadLength. Different portions of the road can be of different types. You are going to generate the road randomly. For example, the following arrays (both arrays should have the same size as the corresponding indexes keep the information about the same portion):

```
roadType = [1, 0, 2, 0, 1, 2]
roadLength = [40, 80, 100, 50, 70, 120]
```

Keeps the following road configuration:



Suppose we have two wheeled vehicles W1 and W2 with positions 150 and 245 respectively. The road type that a vehicle is on is found by finding the gap that includes the current position of the vehicle. For example, W1 is included in the stone road defined in range [120,220), and W2 is included in the asphalt road in range [220,270).

Include an *int getType(int position)* method in your simulation class to find the type of road given the current position of a vehicle. Position variable of the vehicle indicates how far it is from the starting zero-point.

Your simulation class should keep an array of vehicles that are created randomly.

You should get the number of vehicles to create from the user. A newly created vehicle should be wheeled with 40% probability, flying with 25% probability, and quadruped with 35% probability. To simulate this probability, you may generate a random float between 0-1, and check if it is in a certain range.

A wheeled vehicle has 15-25 (these are the ranges) speed with 30-40 fuel. A flying vehicle has 20-30 speed with 20-30 fuel. A quadruped vehicle has 20-40 speed with 10-20 fuel. Name of each vehicle is determined based on their type and order of creation. Suppose we create 3 wheeled, 5 flying, 2 quadruped, and then again 3 wheeled vehicles; their names would be: W1, W2, W3, F1, F2, F3, F4, F5, Q1, Q2, W4, W5, W6, respectively. Use the initial letter of the subclass and its order number for the name.

Include a *void simulate()* method in your simulation class. This method should start and progress the simulation based on the following logic:

- Get total length of the road from user and randomly create different portions of various road types. This would fill the roadType and roadLength arrays.
- Print the road.
- Get the number of vehicles from the user and create the vehicles.
- All vehicles are at position 0 in the beginning.
- At each turn:
  - Print the name, current position, speed (without multiplying with the factor), and fuel of each vehicle.
  - Move each vehicle calling the move method based on its current road type.
  - If a vehicle reaches the end of the road (the position becomes greater than or equal to the total road length), we call it the winner and end the simulation. In this case print out the name of that vehicle and its speed.
  - Repeat the process until there is a winner or all of the vehicles have 0 fuel.

You can include any necessary classes, variables and methods to support the required functionality.

A sample output for the resulting system is as follows:

```
Please enter the road length: 50
```

```
The following road is generated:
```

```
| -Dirt 20- | -Asphalt 15- | -Stone 5- | -Dirt 10- |
```

Please enter vehicle count: 3

The following vehicles are generated:

W1 - Speed: 20 - Fuel: 33

W2 - Speed: 25 - Fuel: 30

F1 - Speed: 20 - Fuel: 20

Turn 1:

W1 - Position: 0 - Speed: 20 - Fuel: 33

W2 - Position: 0 - Speed: 25 - Fuel: 30

F1 - Position: 0 - Speed: 20 - Fuel: 20

Movements:

W1 moves from dirt, for  $20 * 0.75 = 15$  units

W2 moves from dirt, for  $25 * 0.75 = 18.75$  units

F1 moves from dirt, for  $20 * 1 = 20$  units

Turn 2:

W1 - Position: 15 - Speed: 20 - Fuel: 32

W2 - Position: 18.75 - Speed: 25 - Fuel: 29

F1 - Position: 20 - Speed: 20 - Fuel: 19

Movements:

W1 moves from dirt, for  $20 * 0.75 = 15$  units

W2 moves from dirt, for  $25 * 0.75 = 18.75$  units

F1 moves from asphalt, for  $20 * 1 = 20$  units

Turn 3:

W1 - Position: 30 - Speed: 20 - Fuel: 31

W2 - Position: 37.5 - Speed: 25 - Fuel: 28

F1 - Position: 40 - Speed: 20 - Fuel: 18

Movements:

W1 moves from asphalt, for  $20 * 1 = 20$  units

W1 finishes the race! Position: 50 - Speed: 20 - Fuel: 30

End of simulation. Do you want to play again?

**Preliminary Submission:** You will submit an early version of your solution before the final submission. This version should at least include the following:

- You should complete your vehicle, wheeled, flying and quadruped classes. Do not forget to test the methods of these classes in your simulation class.

You will have time to complete your solution after you submit your preliminary solution. You can consult the TAs and tutors during the scheduled Zoom meeting. Do not forget to make your final submission at the end. Even if you finish the assignment in the preliminary submission, you should submit for the final submission on Moodle.

**Not completing the preliminary submission on time results in 50% reduction of this assignment's final grade.**