Functions (name is highlighted using the bolded **Consolas font**). The program feature specified in the requirements document is underlined. Note: not all options are explicitly specified in the requirements but are implicitly required e.g. in order to allow the player to select cheat options such as toggle debugging mode, teleport the car etc. then the program needs a cheat menu to display these options and to react to (process) the option selected by the player.

* Artic track’s abilities:
	+ arctic track can have a blizzard **processBlizzard**: randomly determines if a blizzard has occurred during a turn and then sets the state of the track (Boolean attribute) accordingly.
* Desert track’s abilities:
	+ desert track can be hit with a heat wave **processHeatWave**: randomly determines if a heat wave has occurred during a turn and then sets the state of the track (Boolean attribute) accordingly.
* Track’s abilities:
	+ (Not in requirements but implicitly needed) **display**: shows the current state of the 1D grid
	+ cars reach the end of the track then the simulation ends **isWon**: after both cars have moved each track checks if a car on a particular track has reached the end.
	+ (Not in requirements but implicitly needed) **setLocation**: given an index (location) and a car this function/method will move the car to the specified index (car appears at that location in the grid)
* Basic Car (applies to any car whether sports car or SUV):
	+ Running out of fuel: **consumeFuel**: reduces car’s fuel supply.
	+ car will try to reach the end **Move**: Displays information (current fuel level, consumption rate, distance traveled)
* SportsCar:
	+ (All Wheel Drive) movement: **disEngageAWD(),engageAWD(),** turns the AWD off/on respectively (AWD state is tracked with a Boolean flag and the car will move differently depending upon whether the mode is on or off). **inAWD(),** a Boolean function that queries the state of the AWD flag – needed when the artic track has a blizzard occurring, **moveAWD(),** when the car is in this mode then it will move the specified distance consuming the specified amount of fuel (those details aren’t hugely important for the purposes of this exercise).
* GameController (interface and controls the rules of the simulation)
	+ Display menu options: **displaySUVMenu**, **displaySportsMenu**, 'cheat' option **displayCheatMenu**
	+ Get user selection for menu (handles all three menu types): **getMenuSelection**
	+ **processSportsMenuSelection**: determines which menu option the user selected when controlling the sports car and calls another function as appropriate.
	+ **procesSUVMenuSelection**: determines which menu option the user selected when controlling the SUV and calls another function as appropriate.
	+ 'cheat' option **processCheatMenuSelection**: determines which menu option the user selected when the hidden cheat option is selected and calls another function as appropriate.
	+ car will try to reach the end: **moveCar**: for a particular car on a particular track it moves the car by the amount determined by car type.
	+ **displayTracks**: calls the functions that display each of the two tracks: artic and desert.
	+ **displayArticTrack**: displays the current state of the artic track.
	+ **displayDesertTrack**: displays the current state of the desert track.
	+ cars reach the end of the track then the simulation ends**: determineOutcome**: after both cars move it determines if the race has ended and which car has won.