

2. A train is composed of an engine and any number of train cars. The engine is rated to pull up to a maximum weight, based on its power. The weight of all the train cars combined, including the engine, must be below this maximum weight or the engine will not be able to move the train. The following complete `Engine` class is used to represent the train's engine.

Precondition: The maximum weight rating is greater than or equal to the weight of the `Engine` (every `Engine` object can pull itself).

```
/**
 * Precondition: maximumWeight is greater than or
 *               equal to the weight of the engine.
 */
public class Engine
{
    private double weight, maximumWeight;

    public Engine(double wt, double maxWt)
    {
        weight = wt;
        maximumWeight = maxWt;
    }

    public double getWeight()
    { return weight; }

    public double getMaximumWeight()
    { return maximumWeight; }
}
```

Various train cars are represented by the `TrainCar` class. The weight of a `TrainCar` is its base weight plus the weight of its contents. The base weight of a `TrainCar` is the weight of the empty car. Many types of `TrainCar` objects exist, including passenger cars, boxcars, flatcars, cabooses, and so on. Each of these types has a different way of calculating the weight of its contents. Since each type of `TrainCar` does this differently, the `getTotalWeight` method is an abstract method.

```
public abstract class TrainCar
{
    private double baseWeight;

    public TrainCar(double baseWt)
    { baseWeight = baseWt; }

    public double getBaseWeight()
    { return baseWeight; }

    public abstract double getTotalWeight();
}
```

- (a) `TrainCar` objects that are designed to transport people are called passenger cars and are represented by objects of the `PassengerCar` class. The weight of a `PassengerCar` is calculated by adding the base weight of the `PassengerCar` to the weight of the maximum number of people that the car can hold. The calculation assumes that the passenger car is filled to capacity and that each person, including luggage, weighs 300 pounds.

Write the complete `PassengerCar` class, including a constructor and any necessary instance variables and methods. The constructor takes as parameters the base weight of the car and the maximum number of passengers the car can hold. The class does not need to include a `toString` method or accessors and modifiers for any instance variables.

- (b) Trains are represented by the `Train` class and contain an `Engine` followed by any number of `TrainCar` objects. You may assume that the various classes that extend `TrainCar` have been correctly implemented and are not shown.

The weight of a `Train` object is calculated by adding the weight of the `Engine` to the weight of each of the `TrainCar` objects in the `trainCars` `ArrayList`. The `Train` class contains a `removeExcessTrainCars` method that ensures the engine is able to pull the train (to be completed in part (b)).

```
public class Train
{
    private Engine engine;
    private ArrayList<TrainCar> trainCars;

    public Train(Engine e, ArrayList<TrainCar> tc)
    {
        engine = e;
        trainCars = tc;
    }

    /** Removes TrainCar objects from the end of the train
     * until the train can be pulled by the Engine.
     *
     * @return ArrayList<TrainCar> containing the removed cars in
     * the order they were removed (the last car is
     * item 0, etc.). If no cars are removed, the returned
     * list will be empty.
     */
    public ArrayList<TrainCar> removeExcessTrainCars()
    { /* To be implemented in part (b) */ }

    /* Additional implementation not shown */
}
```

Trains need to be checked to make sure that their weight can be pulled by their `Engine`. If the train is overweight, train operators must remove train cars from the end of the train until the train is within the acceptable weight range.

Write the `removeExcessTrainCars` method of the `Train` class that removes `TrainCar` objects one at a time from the end of the train until the train is less than or equal to the maximum weight allowed as given by the `getMaximumWeight` method of the `Engine` object. The removed train cars are added to the end of an `ArrayList` of `TrainCar` objects as they are removed. This `ArrayList` of removed `TrainCar` objects is returned by the method. If no `TrainCar` objects need to be removed, an empty `ArrayList` is returned.

Example:

A train is composed of the cars listed below. The engine has a maximum weight rating of 475,000 pounds.

Car	Weight
Engine	200,000
<code>trainCars[0]</code>	100,000
<code>trainCars[1]</code>	150,000
<code>trainCars[2]</code>	50,000
<code>trainCars[3]</code>	100,000
<code>trainCars[4]</code>	50,000

In the example, the train initially weighs 650,000 pounds. Train cars need to be removed one by one from the end of the train until the total weight is under the maximum allowed weight of 475,000.

- Removing the train car from index 4 lowers the weight to 600,000
- Removing the train car from index 3 lowers the weight to 500,000
- Removing the train car from index 2 lowers the weight to 450,000

At 450,000 pounds, the train is now in the acceptable weight range. The following `ArrayList` is returned by the method:

```
[trainCars[4], trainCars[3], trainCars[2]]
```

Complete the `removeExcessTrainCars` method.

```
/** Removes TrainCar objects from the end of the train
 * until the train can be pulled by the Engine.
 *
 * @return ArrayList<TrainCar> containing the removed cars in
 * the order they were removed (the first car removed is
 * item 0, etc.) If no cars are removed, the returned
 * list will be empty.
 */
public ArrayList<TrainCar> removeExcessTrainCars()
```