```
#ifndef _TRAIN_H_
#define _TRAIN_H_

// ========================================================================
//
// IMPORTANT NOTE: Do not modify this file
//
// ========================================================================
//
// The TrainCar class is a simple node in a classic doubly-linked list
// data structure.  Each TrainCar has a type (stored as a char), a
// unique ID, a weight, and next & prev pointers to neighboring
// TrainCars.

#include <cstddef>

class TrainCar {

public:
  // static helper functions for construction
  // (the constructor is private)
  static TrainCar* MakeEngine()              { return new TrainCar('e',150);
      }
  static TrainCar* MakeFreightCar(int weight) { return new TrainCar('f',weight);
      }
  static TrainCar* MakePassengerCar()        { return new TrainCar('p',50);
      }
  static TrainCar* MakeDiningCar()           { return new TrainCar('d',50);
      }
  static TrainCar* MakeSleepingCar()         { return new TrainCar('s',50);
      }

  // ACCESSORS
  int getID() const           { return id; }
  int getWeight() const       { return weight; }

  bool isEngine() const       { return (type == 'e'); }
  bool isFreightCar() const   { return (type == 'f'); }
  bool isPassengerCar() const { return (type == 'p'); }
  bool isDiningCar() const    { return (type == 'd'); }
  bool isSleepingCar() const  { return (type == 's'); }

private:
  // private constructor
  TrainCar(char t, int w) : type(t), weight(w), prev(NULL), next(NULL) {
    // each train car has a unique identifer,
    // numbered sequentially, starting at 100
    static int next_id = 100;
    id = next_id;
    next_id++;
  }

  // REPRESENTATION
  // these three member variables cannot be edited after object construction
  int id;
  char type;
```

```
  int weight;
public:
  // these two member variables can be publicly read & edited
  TrainCar* prev;
  TrainCar* next;
};


// ========================================================================
//
// IMPORTANT NOTE: Do not modify this file
//
// ========================================================================

// These helper functions are implemented in main.cpp

void SanityCheck(TrainCar* train);
void PrintTrain(TrainCar* train);

// There are a number of additional functions used in main.cpp that
// you need to declare and implement.  Study main.cpp and the provided
// output to determine the arguments, return type, and overall
// behavior.

// Instead of editing this file directly, place your prototypes for
// additional functions in this file:

#include "traincar_prototypes.h"

// And implement these functions in "traincar.cpp"

// ========================================================================
//
// IMPORTANT NOTE: Do not modify this file
//
// ========================================================================

#endif
```