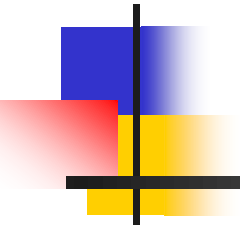


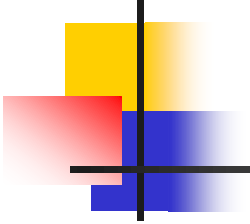
Project 12: Cars from File

Project 12: Cars from File

- This is an extension of Project 11, Car Class
  - You may use the posted solution for Project 11 as a starting point for this project.
  - You might also find the posted solution for Project 10 helpful.
  - It is OK to copy code from any file posted on the class web site.

Assignment

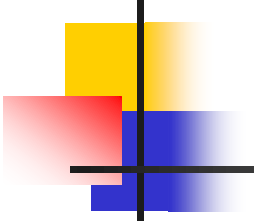
- Add a constructor to the Car class from Project 11 that takes a file Scanner object as its only parameter.
  - The file will consist of attributes of cars
    - As read from the keyboard in Project 11
- Let the new constructor read three lines from the file, using the scanner object passed by the caller, to get the attributes of a single car, and use those values to initialize a Car object:
  - Make
  - Model
  - Year
- Remember that you will need to clear a blank line from the Scanner after reading the year as an integer.



Assignment

Create a test driver called CarsFromFile.

- Accepts input from the keyboard for a file name.
- Creates a Scanner object for the file.
- Repeated invokes the new Car constructor, passing the Scanner object as the only parameter value.
- Adds the resulting Car object to an array.
- Upon reaching end of file, steps through the array of Cars and outputs the attributes of each car to the screen.



CarsFromFile.java

- You may assume that there will be no more than 100 cars in the file.

- If the file specified by the user does not exist, output an error message and let the user try again.

Implementation Tips

- As always, work in tiny steps.

- Start by adding a constructor to the Car class in the posted solution for Project 11: Car.java
  - The new constructor should take a Scanner object as its only parameter.
  - Use the Scanner object to read three lines from the file and initialize the Car member variables with the values from the file.
  - Nothing else in Car.java should be changed.

- Get your modified Car.java to compile.

Step 2

- Write a test driver to test the new constructor with just one car.
- Create a test input file with information for one car.
- Let the test driver
  - Create a File object for your test input file, using a fixed file name.
  - Create a Scanner object using the File object.
  - Create a Car object using the new constructor that takes a Scanner as its parameter.
  - Display the car using println.

Implementation Tips

Step 3

- Extend your test driver to work with an input file having an arbitrary number of cars.
  - (No more than 100.)
- Still use a fixed (hard coded) file name.
- Add more cars to your test input file.
- Build an array of Car objects containing the Cars from the file.
- Upon reaching end of file, step through the array and output the attributes of each Car, using println().

Implementation Tips

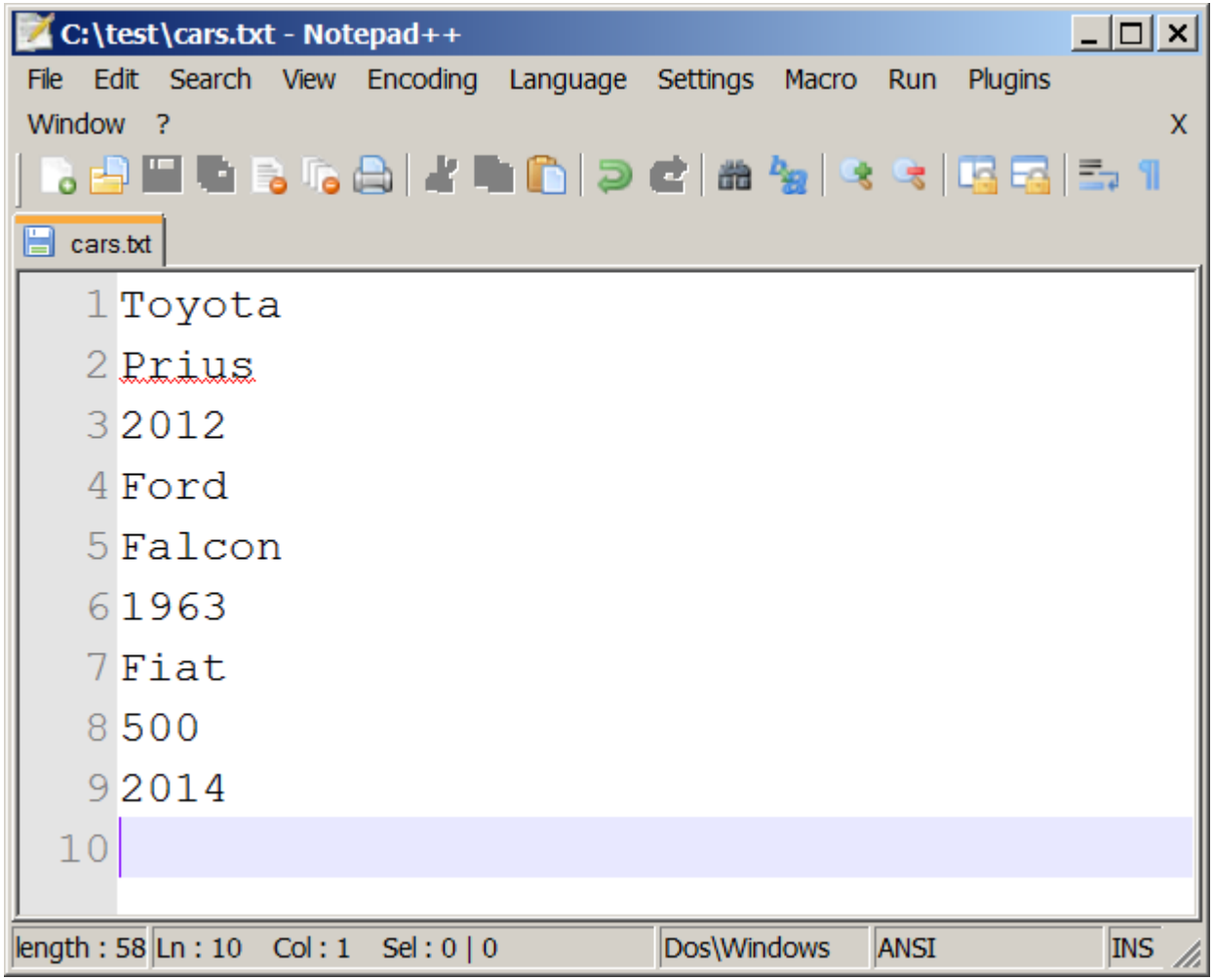
Step 4

- Add code to your test driver to get the file name from the user.
  - Initially let the program crash if the file does not exist.

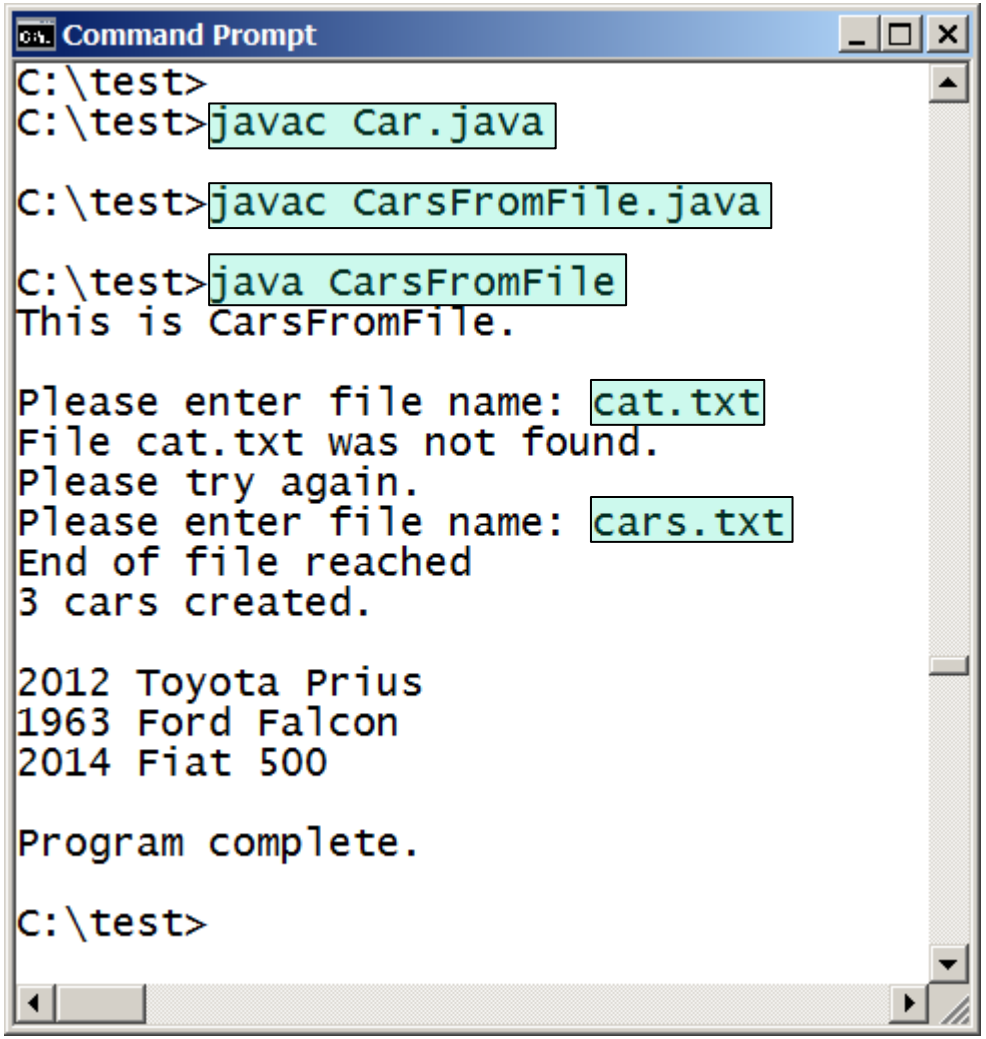
Step 5

- Add code to output an error message if the file specified by the user does not exist and let the user try again.

A Test File


C:\test\cars.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins
Window ?
cars.txt
1 Toyota
2 Prius
3 2012
4 Ford
5 Falcon
6 1963
7 Fiat
8 500
9 2014
10
length : 58 Ln : 10 Col : 1 Sel : 0 | 0
Dos\Windows
ANSI
INS

Sample Run


Command Prompt
C:\test>
C:\test>javac Car.java
C:\test>javac CarsFromFile.java
C:\test>java CarsFromFile
This is CarsFromFile.
Please enter file name: cat.txt
File cat.txt was not found.
Please try again.
Please enter file name: cars.txt
End of file reached
3 cars created.
2012 Toyota Prius
1963 Ford Falcon
2014 Fiat 500
Program complete.
C:\test>

Submission

- Put your Java source files into a folder and zip it.
- Submit your zipped folder via Canvas Assignments.

- Project is due by 11:59 PM
  - Sunday, April 3 Sections 1 and 2
  - Monday, April 4 Sections 3 and 4

- Recommendation:

  Do this project in your lab session or help session.

Ground Rules

- It is OK to *discuss* the project with other students BUT
  - Do not share your code with other students.
  - Before or after submitting the project.
- Do not copy any other student's code.
  - Or even look at it.
- Do not let anyone copy or examine your code.

Ground Rules

Except for code posted on the class web site

- Do not copy code from the Internet
  - or any other source (other than the textbook.)
- Do not ask for help on an Internet forum.
  - If you need help, ask your instructor or a TA.
  - Come to lab and help sessions.
- Write your own code.