

Practice: Creating Classes and Objects

Try these problems, don't just look at the answers. If you look at the answers and find you made mistakes, write the answers out again by hand, even if you are mostly copying them. People learn differently from writing versus reading details: you are much more likely to absorb the pattern of defining classes if you actually write it out yourself. If you type instead of write by hand, don't use cut and paste. You actually have to write/type the characters to help you absorb the patterns.

1 Problems

1. Create a class that captures *students*. Each student has a first name, last name, class year, and major (a shorter term than “concentration”). Create two examples of students.
2. Create a class that captures *planets*. Each planet has a name, a distance from the sun, and its gravity relative to Earth's gravity. For distance and gravity, use the type `double` which captures real numbers. Make objects for Earth and your favorite non-earth planet.
3. Create classes that capture *bank customers* and *bank accounts*. A customer has a first and last name. An account has a customer and a balance. Make objects for two accounts held by the same customer.
4. Create a class that captures *airline tickets*. Each ticket lists the departure and arrival cities, a flight number, and a seat assignment. A seat assignment has both a row and a letter for the seat within the row (such as 12F). Make two examples of tickets.

2 Answers

<https://brown-cs18-master.github.io/content/practice/Practice1.java>

You might have used different names for the fields, but your answers should have been close to these. The `Practice1Test` class is not complete, but just shows you one example of creating data in which one object contains another. Your `Practice1Test` class should be much more extensive.

2.1 Things to Consider About the Answers

- For seat assignments, you could use a string or you could make a seat-assignment class. What are the advantages and disadvantages of each approach?
- Is there any difference between using the same or different objects to represent the same customer across two accounts?

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS18 document by filling out the anonymous feedback form: <https://cs.brown.edu/courses/cs018/feedback>.