

Session 2 - Exercises

Use pen and paper to solve these exercises.

<class-declaration> ::= <visibility-modifier> class <class-name> { <class-body> }

1. Write the class declaration for the below classes, omit the class bodies. All classes should have **public** visibility.

Classes: Car, Customer, Person

2. What pattern do we need to fill in when writing instance variables? Write down the production rule.

3. Declare the following instance variables:

An instance variable that can hold the name of a customer

An instance variable that can hold the age of a customer

An instance variable that can hold the age of a horse

An instance variable that remembers whether the customer wants to receive our newsletter

An instance variable that can contain the name of the city the customer lives in

4. Write the production rule for a constructor (*<constructor> ::= ...*)

5. Write a constructor without parameters for the following classes: Person, Customer, Car

6. Declare a class (class declaration **and** instance variables) that can keep track of a customer, including his or her name, address, email, and year of birth.

7. Write a constructor for the customer class from exercise 6. that always specifies a name and email for the customer object created.

8. Write a constructor for the customer class from exercise 6 that always specifies a name, an email, an address, and the year of birth for the newly created customer object.

9. Open BlueJ, create a new project, and code the Customer class from exercise 6., 7., and 8. Create three customers:

a.) Banana Joe, Tropical Island 7, joe_b@example.com, 1955

b.) Mr. Worf, Klingon Empire, worf@example.com, 2340

c.) you make up some data or insert your own personal info

9.1. Observe how the data is stored, and whether you can change any of the data specified. Use the object inspector to see what data each object holds.

10. Evaluate the following expressions (provide the valuations using arrows) as shown in the slides.

a.) `int x = 7;`

b.) `int x = 7;`

`int y = x;`

c.) `int x = 6 + 7;`

d.) `int x = 2;`

`int y = 6 + x;`

e.) `int x = 6;`

`int y = 1 + x;`

`int z = x + y;`

f.) `int count = 5;`

`count = 2 * count;`

11. Specify the syntactic production rule for getters and setters.

,

12. Open the customer class that you created previously in BlueJ. Add getters and setters for all the attributes.

12.1. Create customer objects Banana Joe, Mr. Worf and the person of your choice. Right click the objects and call some of the setters and getters. Keep the object inspector open for the three objects and observe how the values of the attribute change.

Guru meditation:

In exercise 12.1 YOU are part of the running program. Imagine that you are “the rest of the program” that needs access to customer data. One part of you created the objects previously, therefore they exist now. When you look at the attributes and modify them, you, the “rest of the program” are modifying the data stored in the system.

Understand how and why these objects are useful. Discuss with your peers and the teacher. Ask, challenge.