Exercise 2.2 What value is returned if you check the machine's balance after it has printed a ticket? ZERO

Exercise 2.3 Experiment with inserting different amounts of money before printing tickets. Do you notice anything strange about the machine's behavior? What happens if you insert too much money into the machine – do you receive any refund? NO What happens if you do not insert enough and then try to print a ticket? STILL YOU CAN PRINT THE TICKET

Exercise 2.5 Create another ticket machine for tickets of a different price. Buy a ticket from that machine. Does the printed ticket look different? NO

Exercise 2.6 Write out what you think the outer wrappers of the Student and LabClass classes might look like – do not worry about the inner part.

```
public class Student
{
}

public class LabClass
{
}
```

Exercise 2.7 Does it matter whether we write public class TicketMachine or in the outer wrapper of a class? Edit the source of the TicketMachine class to make the change and then close the editor window. Do you notice a change in the class diagram?
What error message do you get when you now press the *Compile* button?
<Identifier expected>
Invalid method declaration; return type required

Do you think this message clearly explains what is wrong? NO

Exercise 2.8 Check whether or not it is possible to leave out the word public from the outer wrapper of the TicketMachine class.
NO

Exercise 2.9 From your earlier experimentation with the ticket machine objects within BlueJ you can probably remember the names of some of the methods – printTicket, for instance. Look at the class definition in Code 2.1 and use this

knowledge, along with the additional information about ordering we have given you, to try to make a list of the names of the
fields, PRICE, BALANCE, TOTAL
constructors, TICKET MACHINE
and methods in the TicketMachine class. getPrice, getBalance, InsertMoney, printTicket.
Hint: There is only one constructor in the class.

Exercise 2.10 Do you notice any features of the constructor that make it significantly different from the other methods of the class?
THE NAME IS EXACTLY THE SAME AS THE CLASS,
IT DOESN'T HAVE A RETURN OPTION
IT DOESN'T CONTAIN VOID IN THE SIGNATURE
Constructor is called only once at the time of Object creation

Exercise 2.11 What do you think is the *type* of each of the following fields?
private int count; INTEGER
private Student representative; STRING
private Server host; STRING

Exercise 2.12 What are the *names* of the following fields?
private boolean alive; ALIVE
private Person tutor; TUTOR
private Game game; GAME

Exercise 2.13 In the following field declaration from the TicketMachine class
private int price;
does it matter which order the three words appear in? YES Edit the TicketMachine class to try different orderings. After each change, close the editor. Does the appearance of the class diagram after each change give you a clue as to whether or not other orderings are possible? Check by pressing the *Compile* button to see if there is an error message. <IDENTIFIER EXPECTED>
Make sure that you reinstate the original version after your experiments!

Exercise 2.14 Is it always necessary to have a semicolon at the end of a field declaration? YES Once again, experiment via the editor. The rule you will learn here is an important one, so be sure to remember it. ERROR: ; EXPECTED

Exercise 2.15 Write in full the declaration for a field of type int whose name is status.
Int status;

Exercise 2.16 To what class does the following constructor belong? public Student(String name)
STUDENT

Exercise 2.17 How many parameters does the following constructor have and what are their types?
public Book(String title, double price)
2 - - STRING AND DOUBLE

Exercise 2.18 Can you guess what types some of the Book class's fields might be? Can you assume anything about the names of its fields?

Exercise 2.19 Suppose that the class Pet has a field called name that is of type String. Write an assignment statement in the body of the following constructor so that the name field will be initialized with the value of the constructor's parameter.
public Pet(String petsName)
{String name = petsName;
... }

Exercise 2.20 *Challenge exercise* What is wrong with the following version of the constructor of TicketMachine?

```
public TicketMachine(int ticketCost)
{
    int price = ticketCost;
    balance = 0;
    total = 0;
}
```