

Exercises - Week 2

Exercises on class definitions

2.1. To what class does the following constructor belong?

```
public Student(String fullName, String studentID)
```

2.2 How many parameters does the following constructor have, and what are their types? To what class does it belong?

```
public Book(String title, int price)
```

2.3 Do you think the values of the Book constructor's parameters will need to be stored as fields? Can you guess what types some of the class's fields might be, from the parameters to its constructor? Can we know anything for sure about the names of its fields?

2.4 Write a getter method **getTotal** for the class **TicketMaster** to return the value of the **total** field. Code of **TicketMaster** below (fields only):

```
public class TicketMachine
{
    // The price of a ticket from this machine.
    private int price;
    // The amount of money entered by a customer so far.
    private int balance;
    // The total amount of money collected by this machine.
    private int total;

    : rest of code omitted
}
```

2.5 Write out the outer wrapping of a class called **Person**. Remember to include the curly brackets that mark the start and end of the class body, but otherwise leave the body empty.

2.6 Write out definitions for the following fields:

- a field of type **String** called **name**
- a field of type **int** called **age**
- a field of type **String** called **code**
- a field of type **int** called **credits**
- a field of type **boolean** called **applyDiscount**

2.7 Write out the body of a constructor for the class **Person**. The constructor should take two parameters. The first is of type **String** and is called **name**. The second is of type **int** and is called **age**. The first parameter should be used to set the value of a field called **name**, and the second should set a field called **age**. You do not have to include the declarations for the fields, just the text of the constructor.

Hint The body of the constructor should contain two statements, one for each field. **Hint** You will need to use the **this** keyword to refer to the fields of the class.

2.8 Complete the body of the following method, whose purpose is to add the value of its parameter to a field named **score**

```
public void addPoints(int points)
{
    TODO- increase the value of score by points
}
```

2.9 Write a setter method for a field called **age** which is of type **int**. Give the parameter the name **currentAge**.

2.10 Write a setter method for the following field:

```
private boolean alive;
```

Give the method's parameter the name **isAlive**.

Note that you do not need to know how the type differs from the integer type to be able to write this method. Just follow the standard pattern for setter methods as described above.

2.11 Rewrite the above setter method using **this**. (i.e when both the parameter and the field have the same name).

Assignment and variables

2.12 Write a statement that assigns the value of 10 to the variable **x**.

2.13 Write a statement that assigns the value of **x** to the variable **y**.

2.14 Write a statement that assigns the value of **y** to the variable **x**.

2.15 What values do each of the variables **x** and **y** have at this point in the program?

2.16 Given the following code:

```
int x = 10;
int y = 17;
```

Write code that swaps the values of x and y. This code should work no matter what the initial values of x and y are. (i.e. no hard coding of 10 and 17).

(**Hint:** you may need to use a third variable, which you should declare at the top of your code).