

## Programming Assignment 2

### Getting started

This week's project requires the knowledge of only some very basic Java operations on numeric values, but is, nevertheless, not trivial due to its logic. In particular, it focuses on the use of integer division and the remainder (a.k.a. the *mod*, or *%*) operator.

When working on a project always make sure you understand the program requirements first, then think about the algorithm that you will use, write a sketch of it in English and test it, and only after you have thought through the details of it, start working on its implementation in Java.

When working on the program it is important to learn to **develop it gradually by implementing one logical step of the algorithm at a time and testing the program after implementing each step.**

As one CS professor wrote in his advice to the students:

**Never ever use the Freshman Despair Approach:**

1. *Procrastinate;*
2. *Type in all the code;*
3. *Fix all the compile-time errors;*
4. *Fix all the run-time errors;*
5. *Add comments.*

*Unless the program is trivial, this approach will always be too time consuming.*

### Programming Project

**ChangeAndTime:** *Calculate change and time left before a movie*

**worth 10 points**

Write a program that determines how to break the change from money received for a movie ticket and also calculates the time remaining before the start of the movie.

The program should get the following information from the user:

1. the price of the ticket as a decimal number of dollars (e.g. 7.50, 3.45)
2. the amount of money received from a person purchasing the ticket,
3. the start time of the movie in hours and minutes based on the 24-hour clock,
4. the current time in hours and minutes based on the 24-hour clock.

It should compute and output the following:

1. the amount to be returned to the purchaser as a decimal number of dollars,

2. the change that is due to the purchaser, in the form of the number of dollar bills, quarters, dimes, nickels and pennies that must be returned. Use coins of larger denomination whenever possible, i.e. use a quarter, instead of two dimes and a nickel.
3. the time left before the start of the movie, in hours and minutes.

Here a sample interaction ( user input is shown in **boldface**)

```
This program computes the change due to a person, and time left before the start
of a movie.
Please enter the price of the ticket (in dollars):  7.50
Please enter the amount received for the ticket (in dollars):  10
The amount due to the purchaser is 2.50.  The change can be given as 2 dollars, 2
quarters, 0 dimes, 0 nickels and 0 pennies.

When does this movie start (please indicate hours and minutes)?  Hours:  14
Minutes:  20
What time is it now (please indicate hours and minutes)?  Hours:  13
Minutes:  55
You have 0 hours and 25 minutes before the start of the movie.
```

In this project you should not worry about the user entering invalid data (for instance, the current time that is past the start of the movie, or insufficient amount paid for the ticket). Your program will be tested with valid input only.

**Grading.** The grading schema for this project is approximately as follows:

- Your program should at least compile without syntax errors to receive any credit.
- 7 points will be awarded to a program that uses good programming style (to be discussed in class) and correctly computes the amount due, and the breakdown of the change.
- Additional 3 points will be awarded for the correct computation of the time remaining before the start of the movie.

Remember that the best way to develop a program is by working on it incrementally and periodically verifying the correctness of each developed part. For instance, after developing the code that computes the amount due, and the breakdown of the change, test it on various inputs to make sure that this part works correctly, and after that, proceed with computing the time left.

*Hint:* The easiest way of computing the change is to convert the amount due to the person into cents (e.g. \$2.50, is 250 cents) and work with this representation. A similar trick works for the computation of the time left.

**Important Note:** Your program will be tested by a computer program before I evaluate it. The tester program is not intelligent enough to interpret the output line and deduce which part of it represents the times and which represents the change. **Therefore, the input and output should appear in the exact order that is shown in the sample interaction.**