

## Practice Assignment 2

Save the source code, test cases and related test report as per question number. Make a zip folder of all the solutions and upload in Moodle.

### **Name\_StudentID\_CSE453\_LAB2**

1. You need to develop a software based on Test Driven Development (TDD). The details of the software you need to implement as follows.

The software will check that the Personal Numbers are constructed as per following rules or not.

Let assume, the personal number consist of 10-digit number: YYMMDD-XYZC.

- The first six digits represent the date of birth with year, month and day, for example 640823, which is 23rd of August 1964.
- The following three digits, XYZ, is a serial number, where Z represents the person's gender. If the person is a female Z is an even number and odd if the person is male.
- The last digit, C, is a checksum and is calculated in the following manner:
  - Multiply the digits in the date and serial number with 2, 1, 2, 1,...

For example:

6 4 0 8 2 3 – 3 2 3

2 1 2 1 2 1 2 1 2

12, 4, 0, 8, 4, 3, 6, 2, 6

- Add the resulting digits:  $1+2+4+0+8+4+3+6+2+6=36$ 
  - If a multiplication result is larger than 10, it becomes the sum of the digits, for example 12  
→  $1+2$
- If the summation [in this case it is 36] is greater than 10, take the last digit from the sum (the 6 in the above example) and subtract it from 10.  
  
In the example it will be  $10-6 = 4$ , the resulting number is the check sum C.
- If the result is 10 then the checksum C is 0.
- If the result is less than 10, then the checksum C will be the digit itself.

The Personal number class shall have the following methods:

- getDate() should return the date of birth, YYMMDD in some form.
- getYear() only return the year

- getMonth() only return the month
- getSex() return if the person is male or female (returning just Z is not allowed).
- getChecksum() return the checksum digit
- When instantiating a personal number class, give the personal number to the constructor as a parameter.
- It shall not be possible to instantiate an invalid personal number. If the instantiation fails an exception shall be thrown.

You are free to choose the types (integers, strings etc.) for your class by yourself.

If needed, you can add more member functions

At least you should have the following test cases for a class that represents the personal number:

1. A test case for checking the accuracy of checksum calculation
2. A test case for getDate() method
3. A test case for getYear() method
4. A test case for getMonth() method
5. A test case for getSex() method
6. A test case to check that that personal number is constructed as per the rules.
7. A test case to check thrown of exceptions if the personal number is not constructed as per the rules.

You should follow TDD. You should have the following steps in your development:

1. Add a test that fails
2. Make the code work
3. Run the test to see if it passes
4. Modify and revise the code (Refactoring)

**60 Points**

2. You are given a template for the Currency and Money classes. For these two classes, at first, write test cases for the methods of each class, then you have to add code to the methods of these classes (You need to follow Test Driven Development –TDD).

The Bank and Account classes were written by a bad programmer. When you are confident that your Money and Currency classes work as intended, write test cases for the Bank and Account classes and find the bugs.

**40 Points**