OBJ2000 Exam – February 2021

Task 1 (10 points)

Write a method that takes a integer value (x) as an argument and returns smallest odd number bigger than input number x.
For exemple:
For x= 25 the method returns 27
For x= 34 the method returns 35
For x= -7 the method returns -5

Task 2 (15 points)

Write Java program that writes the following chessboard using two for loops:

```
A1 B1 C1 D1 E1 F1 G1 H1
A2 B2 C2 D2 E2 F2 G2 H2
A3 B3 C3 D3 E3 F3 G3 H3
A4 B4 C4 D4 E4 F4 G4 H4
A5 B5 C5 D5 E5 F5 G5 H5
A6 B6 C6 D6 E6 F6 G6 H6
A7 B7 C7 D7 E7 F7 G7 H7
A8 B8 C8 D8 E8 F8 G8 H8
```

Task 3 (15 points)

Write a method that takes integer number n as an argument. After that, it checks if n is odd or even. In the case its even, the program should use a while loop to write 5 subsequent even numbers. In the case it is odd, the program should multiply number n with 25, and write a result.
E.g. 1: for n = 6, the program should write: 8,10,12,14,16
E.g. 2: for n = 7, the program should write: 175

Task 4 (10 points)
Write a Java program that transforms the temperature (input from user) in Celsius into temperature in Fahrenheit, using the formula f=9*c/5+32
For example:
if c is 0, the method returns 32
if c is 10, the method returns 50

Task 5 (15 points)
Write a Java program that shows a message about number of digits and letters in a String using the method:

```
public void printNumberOfDigitsAndLetters(String input){}
```

E.g. 1: if the input is: “I am 25 years old”, the method should print:

```
The line contains 2 digits and 11 letters!
```

Task 6 (35 points)

The task is to create a Java program that supports the classification of employees kept in an archive of a university.

1. Create a class named *Employees* that contains data fields for the *employee number*, first name, last name, and age. Include *Get* and *Set* methods for these fields (4 points).
2. Class *Employees* should have two method constructors that assign values to the specific fields. One constructor with three arguments (*number, firstName, lastName*) and one with four arguments (*number, firstName, lastName, age*) (3 points).
3. Class Employees should offer *print* method that writes a message about the first name, last name and age of the employee (3 points).
4. Create two subclasses that will extend class *Employees* and which will be named: *Professors* and *Advisors* (4 points).
5. *Professors* subclass should contain additional fields (1 point):
    a. One that holds a code of the *course* professor teaches.
    b. One field that indicates whether the course is *active* or not.
6. *Professors* should have additional constructor that updates those two additional fields (3 points) in addition to the constructor from the superclass.
7. Add methods *getCourse* and *isActive* that read and return values of the fields *course* and *active*, respectively (3 points).
8. Override the print method from the superclass to create method in *Professors* class to print information from all available fields (3 points).
9. *Advisors* subclass should contain additional field that stores the name of the *department* (that advisor is employed by) (2 points).
10. Use polymorphism to create two methods in *Advisors* class to print information about the advisors. One method should print only first name and last name. Other method should print all available information. (4 points).
11. Create simple *Test* class with main method, that will create instances of one Professor and one Advisor and print details from both (5 points).

Pay attention on the use of access modifiers and readability of the code. During the assessment of this task, special attention will be focused on the ability to implement basic principles of object-oriented programming: inheritance, encapsulation, and polymorphism.