

Exam

DIT948: Programming

Date:	2016-10-27
Time	14:00–18:00
Place	Lindholmen
Teacher	Musard Balliu
Examiner	Ivica Crnkovic
Questions	0729744942 first visit around 15:00
Results	will be posted within 15 working days
Grades	Pass (G) 50p, Pass with Honors (VG) 80p
Allowed aids	Any book on Java programming is allowed Any English dictionary No electronic translators are allowed
Reviewing:	2015-11-21 and 23, 13:00–15:00, Lindholmen Second floor room in Patricia (Note: subject to change, changes will be posted on the course website).

Please observe carefully the following:

- Write in legible English (illegible translates to “no points”!).
- Motivate your answers, and clearly state any assumptions made.
- Code or text copied from the book will not be rewarded. Your own contribution is required.
- Start each task on a new sheet!
- Write on only one side of the paper!
- Before handing in your exam, number and sort the sheets in task order!
- Write your anonymous code and page number on every page!

Not following these instructions will result in the deduction of points!

1. [10pts] Consider the following Java program:

```
public class EasyCrypt {
    private static int key=10;
    private int message;

    public EasyCrypt(int message){
        this.message = message;
    }
    public int encrypt(){
        this.message = this.message * EasyCrypt.key;
        System.out.println("Encrypted message is " + this.message);
        return this.message;
    }
    public int decrypt(){
        this.message = this.message/(EasyCrypt.key);
        System.out.println("Decrypted message is " + this.message);
        return this.message;
    }
    public static void mix(EasyCrypt e1, EasyCrypt e2){
        int m = e1.encrypt();
        e2=e1;
        System.out.println("Encrypted message in mix is " + m);
        EasyCrypt.key = 5;
        int d = e2.decrypt();
        System.out.println("Decrypted message in mix is " + d);
    }
    public static void main(String[] args) {
        EasyCrypt o1 = new EasyCrypt(20);
        EasyCrypt o2 = new EasyCrypt(30);
        o1.encrypt();
        o1.decrypt();
        mix(o1,o2);
        o1.encrypt();
        o1.decrypt();
        o2.encrypt();
        o2.decrypt();
    }
}
```

What will be printed to the console when running EasyCrypt?

2. [10pts]

Your boss wrote a program that was supposed to compute the maximum and the minimum (1-digit) number from a string of digits (string of characters 0, 1, .. 9). The method should be static and return an array of two integers containing the minimum number in the first position and the maximum number in the second position. If the input string is empty, the method should return an array containing 0 both for the minimum and the maximum number.

Your boss used the following APIs in the implementation:

`s.charAt(int i)` that returns the character at position i in string s .
`Integer.parseInt(String s)` that returns the integer corresponding to string s .

Unfortunately, things didn't really work out as expected: in fact, the program doesn't even compile! Still, it's your boss, you can't just throw it away and write a new program. Make the changes necessary for the program to compile and return the correct result.

```
public static []int minMax(String s){
    if(s.length()==0) {
        return new int[]{0,0};
    }
    int[] res = new int[2];
    res[0] = Integer.parseInt(s.charAt(0)+"");
    res[1] = Integer.parseInt(s.charAt(0)+"");

    for (int i=0; i <s.length(); i++){
        int curr = Integer.parseInt(s.charAt(i));
        if (curr==res[1]) {curr = res[1];}
        if (curr < res[0]) {res[0] = curr;}
        return res;
    }
}
```

3. [10pts] Consider the following classes:

```
public class A {
    private String msg;

    public A(String msg){
```

```
        this.msg = msg;
        System.out.println("Constructor class A: " + this.msg);
    }
    public String printMsg(){
        return "Message from class A: "+this.msg;
    }
}
```

```
public class B extends A {
    public String msg1;

    public B(String msg) {
        super(msg);
        this.msg1=msg;
        System.out.println(super.printMsg());
    }
    public String printMsg(){
        return "Message from class B: " + this.msg1;
    }
}
```

Which of the lines 3–8 in class `Test`, if any, will cause a compilation error and why? Comment out such lines (if any) and write what will be printed out to the console when running the class `Test`.

```
0 import static dit948.SimpleIO.*;

1 public class Test {
2     public static void main(String[] args){
3         A a1 = new A("Work Hard!");
4         println(a1.printMsg());
5         A b1 = new B("But don't forget to have some fun!");
6         println(b1.printMsg());
7         B b2 = new B();
8         println(b2.printMsg());
9     }
10 }
```

4. [30pts]

In this exercise, you will implement two classes to represent a word as specified in a dictionary, and a list of words to represent a (very) simplified version of the English dictionary. The classes should provide different variables and methods to perform operations in a dictionary, for example, searching for a given word, printing the list of words, and so on.

Concretely, you will implement the following classes:

A `Word` is specified by the word itself, the description of the word and the language code. For instance, the English word *Hello* can be modelled by an object of class `Word`, where `word="Hello"`, `description="A formal greeting!"` and `language='E'` (here the character 'E' represents the language code for English).

A `EnglishWords` extends a `Word` by (additionally) specifying an array of `Word` objects, representing the dictionary, and a single `Word` object.

You will implement the class `Word` and its subclass `EnglishWords` by “filling in” the data and method definitions. You don’t need to do any error checking or write comments.

Please do not write “between the lines”! On your paper, clearly mark out which code is supposed to go where. For example, in the code supposed to fill in the block A, the instance variable of the class `Word`, should appear as

```
+-----A-----+
|  private String word;|
+-----+
```

and not squeezed on the page with the exam subjects!

Remarks:

- Since block A has been filled in for you above, there are twelve blocks left for you to fill in for this exercise. Blocks B, C and E are worth one point each, blocks D, F, G, and I are worth two points each, blocks H, J and K are worth 3 points each, and blocks L and M are worth 5 point each.

End of remarks.

```
/**
 * This is a class representing a Word in a dictionary
 * A Word consists of a word, a description and a language code
 */
public class Word {
    // private instance variable declaration
    // Block A
    // word, a String representing the word

    //Block B
    // description, a String describing the word
    // language, a char representing the language code

    /**
     * Constructor with parameters
     * It initializes this object with formal parameters below
     * @param word: the String representing the word
     * @param language: the char representing the language code
     * @param description: the String describing the word
     */
    public Word(String word, char language, String description){
        // Block C
    }

    /**
     * Constructor with parameters
     * It initializes this object with formal parameters below
     * @param word, a Word object
     */
    public Word(Word word){
        // Block D
    }

    /**
     * getter method
     * @return word of this object
     */
    public String getWord(){
        // Block E
    }
}
```

```
/**
 * Checks if this Word is the same as Word w, that is,
 * whether or not the word and the language code are the same.
 * @param w: a Word object
 * @return true or false
 */
public boolean sameWord(Word w){
// Block F
}

/**
 * Calculate the length of a given Word, namely
 * the number of characters in the String word
 * @param w: the Word object
 * @return the length of the word in w
 */
public int getLength(Word w){
// Block G
}

/**
 * Returns the string representation of this object.
 * Example: If language is 'E', word is "Hello" and
 * description is "A formal greeting" is returns the String
 * "Language: E, Word: Hello, Description: A formal greeting!"
 */
public String toString(){
// Block H
}
}

/**
 * This is a subclass of class Word containing an array
 * of Words englishWords and a Word wordObj
 */
public class EnglishWords extends Word {
// private instance variable declaration
// englishWords, an array of Word objects
// wordObj, a Word object
// Block I
}
```



```
/**
 * Default constructor of the class
 * It initializes: word with "Hello", language with 'E',
 * description with "A formal greeting", wordObj with the Word object
 * initialized with the 3 values above, and englishWords with an
 * array of 100 words containing nothing (null).
 */
public EnglishWords(){
// Block J
}

/**
 * Constructor with parameters
 * It initializes this object with formal parameters below
 * @param word: a Word object
 * @param englishWords: an array of Words
 */
public EnglishWords(Word word, Word[] englishWords) {
// Block K
}

/**
 * A static method to check whether or not a Word w is
 * in the array of Words ew. Namely, whether there exists
 * a Word with the same word field and the same language code.
 * @param w: the Word to search for
 * @param ew: the array of Words
 * @return true/false
 */
public static boolean find(Word w, Word[] ew){
    Block L
}

/**
 * Returns the list of all Words in the array englishWords
 * Example: "Language: E, Word: Hello, Description: A formal greeting!
 *          Language: E, Word: Hi, Description: An informal greeting!"
 * @return the String representing all Words
 */
public String printWordList(){
```

```
// Block M
}
}
```

5. [20pts] In this exercise, we refer to the classes from above, but do not require you to have implemented them (only that you understand how to create and use instances of them).

- (a) In class `EnglishWords`, implement a method with the signature

```
/**
 * Returns the first word with the maximum number of characters
 * (namely, the longest) in the array englishWords.
 * Assume that the array contains at least one element
 * @return the longest word in the array
 */
public String findLongest()
```

that returns the first word (a `String`) with the maximum number of characters in the array `englishWords`. You can assume that the array contains at least one element.

- (b) In class `EnglishWords`, implement a method with the signature

```
/**
 * A static method that merges two arrays of Word without repetitions.
 * You can assume that arr1 and arr2 do NOT contain repetitions.
 * You are NOT allowed to use ArrayList or other Java classes/methods.
 * @param arr1: the first array without repeated words
 * @param arr2: the second array without repeated words
 * @return the merged array of Word with no repetitions
 */
public static Word[] merge (Word[] arr1, Word[] arr2)
```

that returns a merged array of `Word` objects without repetitions (meaning that the `String` word of a `Word` object should only appear once in the returned array). You can assume that `arr1` and `arr2` do NOT contain repetitions. You are only allowed to use methods from the classes in Exercise 3. Hint: Remember the method `freeVariables()` from Assignment 3?

Each method is worth 10 points.

6. [20pts] (From *CodingBat*)

Write a subroutine/method takes an array of integers and returns true if the value 3 appears in the array exactly 3 times, and no 3's are next to each other. The method should have the following signature:

```
public boolean haveThree(int[] nums)
```

Remember to check the corner cases!

Examples:

```
haveThree({3, 1, 3, 1, 3}) -> true
haveThree({3, 1, 3, 3}) -> false
haveThree({3, 4, 3, 3, 4}) -> false
haveThree({1, 3, 1, 3, 1, 3}) -> true
haveThree({3, 4, 3, 4, 3, 4, 4}) -> true
haveThree({3, 3, 3}) -> false
haveThree({1, 3}) -> false
haveThree({3}) -> false
```