UNIVERSITY
OF
JOHANNESBURG

Computer Science 2A

Extra Assignment 01

2016-03-24

Time: Deadline — 2016-04-05 12h00 Marks: 50

This practical assignment must be uploaded to eve.uj.ac.za **before** 2016-04-05 12h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of JavaDoc comments. Marks will be deducted if these are not present. Every submission **must** include a batch file. See the reminder page for more details.

The Java Development Kit (JDK) has been installed on the laboratory computers along with the Eclipse Integrated Development Environment (IDE).

This practical aims to solidify your understanding of basic classes and GUI interfaces.

This practical is **OPTIONAL**. Start with a new empty project for your solution.

The Department of Defence of Utopia is under attack from the country of Dystopia. Spies at the department need a way to communicate with each other without anyone else being able to understand their messages. The Utopian spymaster has tasked you with creating a Java GUI application to enable spies to encrypt and decrypt messages.

Create a class called **ApplicationFrame** which extends **JFrame**. **ApplicationFrame** will contain a single customised **JPanel**. Create a classed called **CipherPanel** which extends **JPanel**. **CipherPanel** will host all of the UI components which the spy will interact with.

The **CipherPanel** will contain two text areas and two buttons. The first text area will be for a plain text message (not encrypted text) and the second text area will be for cipher text messages (encrypted text). The buttons will be used for encryption and decryption respectively. When spy clicks on the encryption button then the plain text message must be encrypted and displayed in the cipher text area. When spy clicks on the decryption button then the cipher message must be decrypted and displayed in the plain text area.

The **Cipher** class has been provided for you to work with. The **Cipher** class will allow you to encrypt and decrypt a message. You may not make any changes to the **Cipher** class!

- *Hint:* Use the JavaDoc and inline comments to aid in your understanding of how the **`Cipher`** class operates.
- *Hint:* The ***`transform()`*** method can perform both the encryption of plain text as well as the decryption of cipher text.

The **`Main`** class will create a **`ApplicationFrame`** and set the required attributes display the frame.

Mark sheet

1. **`ApplicationFrame`** extends JFrame. [01]
2. **`CipherPanel`**
   (a) Extends JPanel. [01]
   (b) Contains two text areas. [02]
   (c) Contains two buttons. [02]
   (d) Use **`Cipher`** class [04]
   (e) Button click encrypts message. [05]
   (f) Button click decrypts message. [05]
3. Packages [05]
4. Coding convention (structure, layout, OO design) and commenting (normal and JavaDoc commenting). [10]
5. Correct execution. [15]

NB

Submissions which **do not compile** will be capped at 40%!

Execution marks are awarded for a correctly functioning application and not for having some related code.

Reminder

Your submission must follow the naming convention as set out in the general learning guide.

```
SURNAME_INITIALS_STUDENTNUMBER_SUBJECTCODE_YEAR_PRACTICALNUMBER
```

Example

Surname	Berners-Lee
Initials	TJ
Student number	209912345
Module Code	CSC2A10
Current Year	2016
Practical number	P00

```
Berners-Lee_TJ_209912345_CSC2A10_2016_P00
```

Your submission must include the following folders:

- `bin` - *(Required)* Should be empty at submission but will contain runnable binaries when your submission is compiled.
- `docs` - *(Required)* Contains the batch file to compile your solution, UML diagrams, and any additional documentation files. Do not include generated JavaDoc.
- `src` - *(Required)* Contains all relevant source code. Source code must be places in relevant sub-packages!
- `data` - *(Optional)* Contains all data files needed to run your solution.
- `lib` - *(Optional)* Contains all libraries needed to compile your solution.

NB

Every submission **must** include a batch file. This batch files must contain commands which will compile your Java application source code, compile the associated application JavaDoc and run the application. **Do not** include generated JavaDoc in your submission. All of the classes/methods which were created/updated need to have JavaDoc comments.