

Exercise 1

1 Objective

Practice Java classes.

Practice svn.

2 Marking

This exercise will be graded out of 10 marks, where each of the required methods will be graded out of two marks:

1 mark for correctness

1 mark for quality of the code

Up to 2 marks may be deducted (allover) for incorrect folder structure/missing files/additional files.

It is worth 1% of your final mark.

The submission deadline is Jun 3, 2016, 11:50pm.

This is an individual exercise.

Late submission policy: No late submissions are accepted.

3 How to submit your work

1. Your individual svn repository now contains a new directory called E1. Check it out using svn in your local machine. Please note if your CDF user name is c5jsmith, and your local folder is /home/johnny, the absolute pathname of your E1 folder is /home/johnny/c5jsmith/E1.
2. Start eclipse and point your workspace to c5jsmith. Create a new Java project called E1.
3. Create a new package called e1.
4. Within package e1, create a new Java class called GaussInteger and paste the starter code (provided on portal) in it.
5. Complete Java class GaussInteger.java, so that it obeys the specifications below and the descriptions in the starter code.
6. Create a Java class called TestGaussInteger that has the main method. Make sure you test everything that needs to be tested before submitting.
7. From the folder c5jsmith issue the command `svn add E1/src`. This way you make sure you are submitting the source code only.
8. Issue your `svn commit -m "some message ..."`
9. If for any reason you will see other files/folders submitted, please delete them using `svn delete name` and the issue a `commit`.
10. Note: an alternative way to eliminate accidental submission of unwanted files/folders is the use of `svnignore`. If so desired, Google it and use it (on your own!)

4 Specifications for GaussInteger.java

Your task is to complete the implementation of the class `GaussInteger`. A `GaussInteger` object is really a complex number where both real and imaginary part are integers. As such, it does have two properties:

```
int real, imag;
```

Also, the `GaussInteger` class must have two constructors, the default constructor is already provided in the starter code, and the other constructor signature must be:

```
GaussInteger(int real, int imag)
```

The required methods are:

```
public GaussInteger add(GaussInteger operand) - adds the GaussInteger object named operand to the current object. The rule of the addition is the usual one: the real part is added to the real part, the imaginary part is added to the imaginary part. For example, if the string representation of gint1 is 1+2i and the string representation of gint2 is 2-i, the statement gint1.add(gint2) must return the value that has the following representation: 3+i.
```

```
public int moduleSquared() returns the sum of squares of the real part and imaginary part, for example if gint1=1-2i the statement gint1.moduleSquared() must return 5.
```

```
public boolean isInvertible() returns true if and only if the moduleSquared() returns 1.
```

Override the `toString` method so that it produces the usual representation of a complex number.

Override the `equals` method so it guarantees two Gauss integers are equal if and only if their real and imaginary parts match.

5 Checklist

Have you...

- tested your code on the lab computers using **Java 1.8**?
- committed the correct files in the correct directory?
- verified that your changes were committed using `svn list` and `svn status`?