Automated Feedback and Assessment of Programming Exercises

LT&SE Awards Showcase

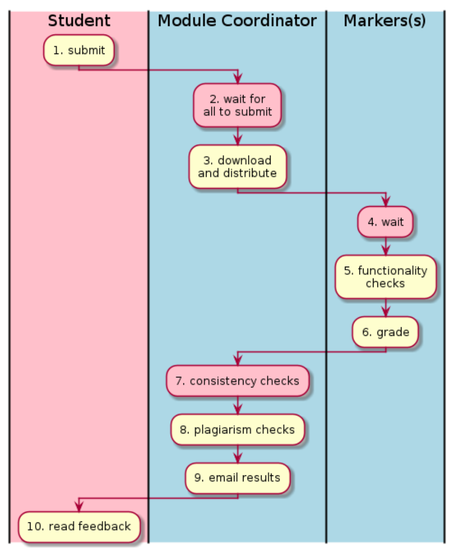
Patrick Totzke
totzke@liverpool.ac.uk

Rahul Savani
rahul.savani@liverpool.ac.uk

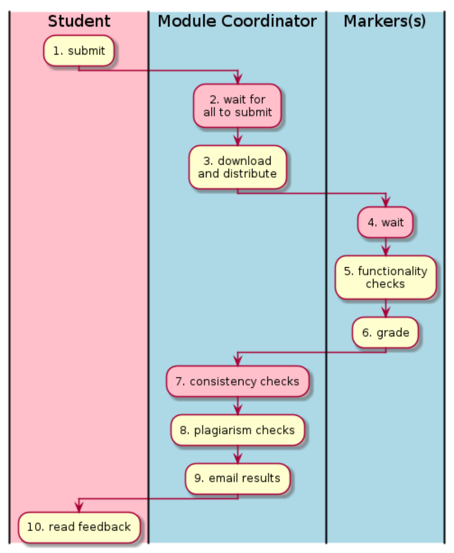
UNIVERSITY OF LIVERPOOL

What’s the Problem?

Assessment of Programming Assignments



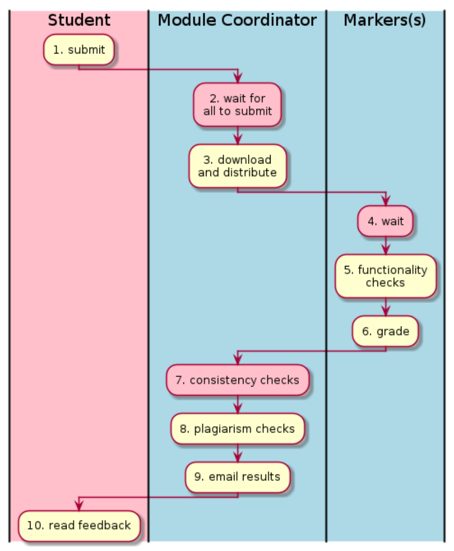
Assessment of Programming Assignments



- **Slow!**

  Taking full 3 weeks is the norm.

Assessment of Programming Assignments



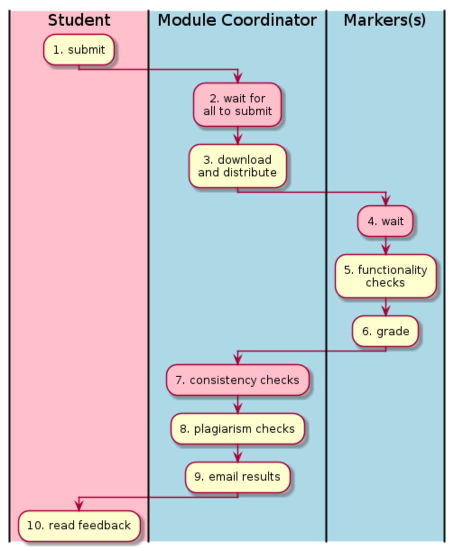
- **Slow!**

  Taking full 3 weeks is the norm.

- **Unfair!**

  No two markers are equally lenient.

Assessment of Programming Assignments



- **Slow!**

  Taking full 3 weeks is the norm.

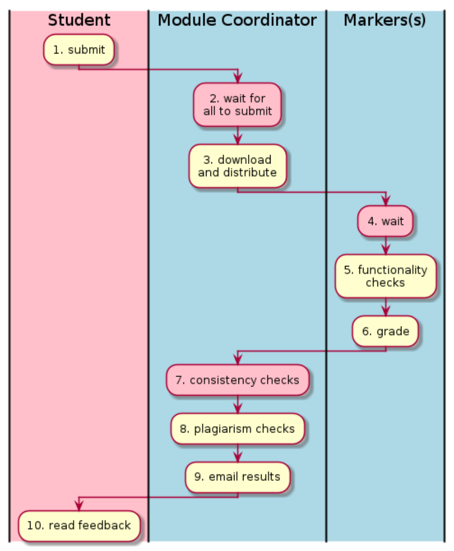
- **Unfair!**

  No two markers are equally lenient.

- **Inconsistent!**

  Experience/quality of marking is not the same across modules;

Assessment of Programming Assignments



- **Slow!**

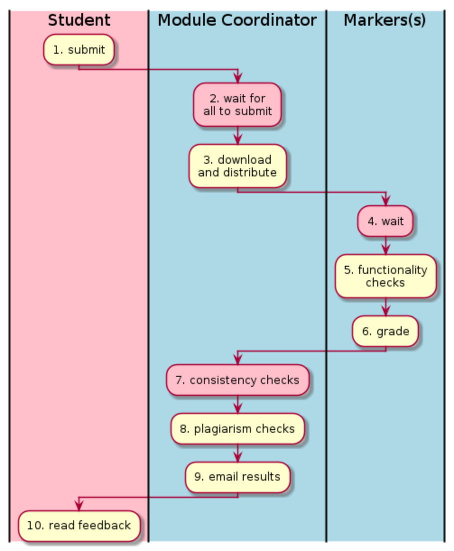
  Taking full 3 weeks is the norm.
- **Unfair!**

  No two markers are equally lenient.
- **Inconsistent!**

  Experience/quality of marking is not the same across modules;
- **lots of work**

  for module coordinator and markers

Assessment of Programming Assignments



- **Slow!**

  Taking full 3 weeks is the norm.
- **Unfair!**

  No two markers are equally lenient.
- **Inconsistent!**

  Experience/quality of marking is not the same across modules;
- **lots of work**

  for module coordinator and markers

. . . should really be automated!

Wishlist: Automated checks should be ...

- available (in part) to students before the deadline
- easy to run and reliable
- flexible in terms of language support
- easy to set up
- cost efficient

Our Activities

Evaluating Software Designs/Solutions

	pre-deadline checks	easy to run	language agnostic	easy to set up	docs & support	cost efficient
own design	🟧	🟧	🟩	🟥	🟥	🟧
Turnitin	🟩	🟩	🟩	🟥	🟧	🟧
Check50	🟩	🟧	🟩	🟧	🟧	🟩
CodeGrade	🟩	🟩	🟩	🟩	🟩	🟧

20/21 L&T Enhancement project

- trial Check50 and CodeGrade (COMP122 and COMP226)
- paid for TAs to implement exercises and software licence

21/22 Extended Trial on CodeGrade

- trial CodeGrade across CS
- 8 modules (CS) + one in Geographic Data Science

Best Option for UoL – CodeGrade

- **available (in part) to students before the deadline?**

Best Option for UoL – CodeGrade

✓ **available (in part) to students before the deadline**

yes, with hidden full checks

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- **easy to run and reliable?**

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- ✓ **easy to run and reliable**

  yes, via web interface with Canvas integration

Best Option for UoL – CodeGrade

✓ **available (in part) to students before the deadline**

yes, with hidden full checks

✓ **easy to run and reliable**

yes, via web interface with Canvas integration

- **flexible in terms of language support?**

Best Option for UoL – CodeGrade

✓ **available (in part) to students before the deadline**

yes, with hidden full checks

✓ **easy to run and reliable**

yes, via web interface with Canvas integration

✓ **flexible in terms of language support**

has full access to a virtual linux machine

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- ✓ **easy to run and reliable**

  yes, via web interface with Canvas integration

- ✓ **flexible in terms of language support**

  has full access to a virtual linux machine

- **easy to set up, docs and support available?**

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- ✓ **easy to run and reliable**

  yes, via web interface with Canvas integration

- ✓ **flexible in terms of language support**

  has full access to a virtual linux machine

- ✓ **easy to set up, docs and support available**

  has extensive docs + 24/7 tech support

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- ✓ **easy to run and reliable**

  yes, via web interface with Canvas integration

- ✓ **flexible in terms of language support**

  has full access to a virtual linux machine

- ✓ **easy to set up, docs and support available**

  has extensive docs + 24/7 tech support

- **cost efficient?**

Best Option for UoL – CodeGrade

- ✓ **available (in part) to students before the deadline**

  yes, with hidden full checks

- ✓ **easy to run and reliable**

  yes, via web interface with Canvas integration

- ✓ **flexible in terms of language support**

  has full access to a virtual linux machine

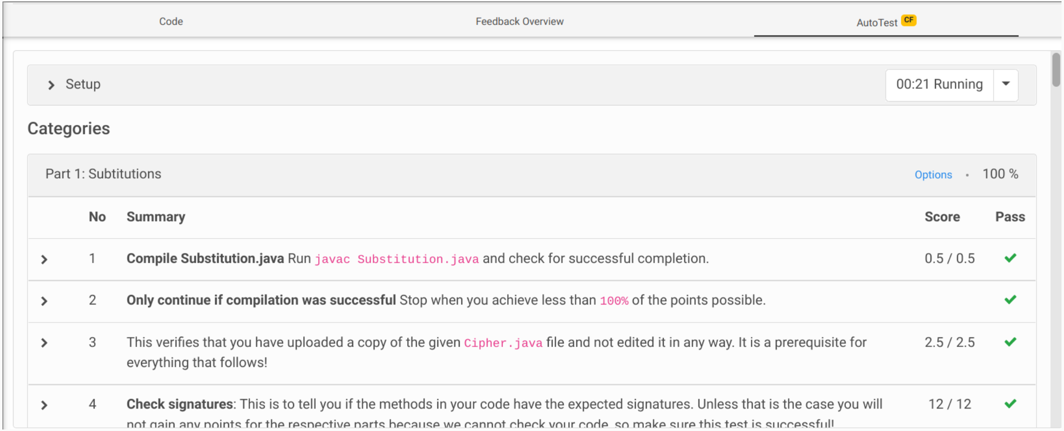
- ✓ **easy to set up, docs and support available**

  has extensive docs + 24/7 tech support

- ✓ **cost efficient**

  Yes, considering cost of licence and one-off setup costs to MC
  vs TA time saved (≈ 15mins per student and assignment)

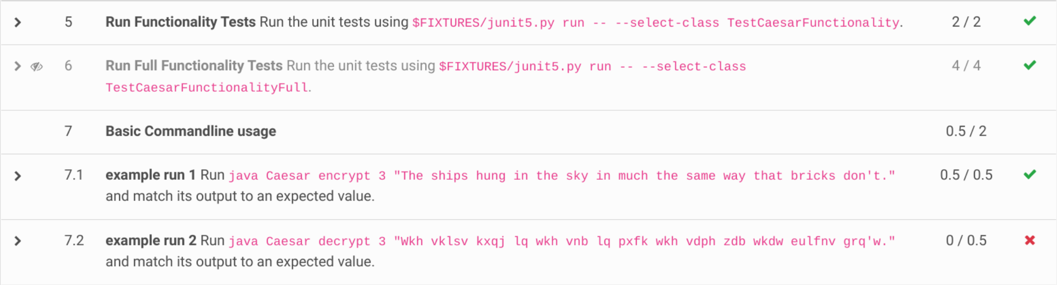
Demo CodeGrade features?



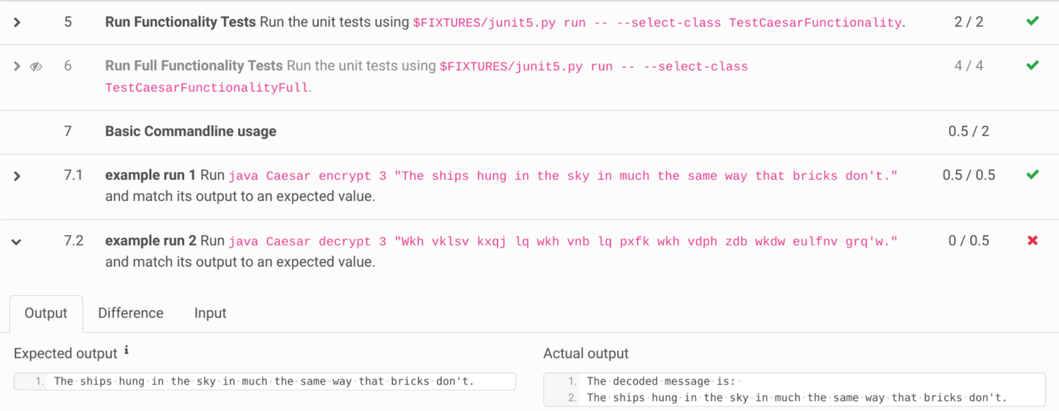
Categories

Part 1: Subtitutions

No	Summary	Score	Pass
1	**Compile Substitution.java** Run `javac Substitution.java` and check for successful completion.	0.5 / 0.5	✓
2	**Only continue if compilation was successful** Stop when you achieve less than `100%` of the points possible.		✓
3	This verifies that you have uploaded a copy of the given `Cipher.java` file and not edited it in any way. It is a prerequisite for everything that follows!	2.5 / 2.5	✓
4	**Check signatures**: This is to tell you if the methods in your code have the expected signatures. Unless that is the case you will not gain any points for the respective parts because we cannot check your code, so make sure this test is successful!	12 / 12	✓



5	**Run Functionality Tests** Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionality`.	2 / 2	✓
6	**Run Full Functionality Tests** Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionalityFull`.	4 / 4	✓
7	**Basic Commandline usage**	0.5 / 2	
7.1	**example run 1** Run `java Caesar encrypt 3 "The ships hung in the sky in much the same way that bricks don't."` and match its output to an expected value.	0.5 / 0.5	✓
7.2	**example run 2** Run `java Caesar decrypt 3 "Wkh vklsv kxqj lq wkh vnb lq pxfk wkh vdph zdb wkdw eulfnv grq'w."` and match its output to an expected value.	0 / 0.5	✗



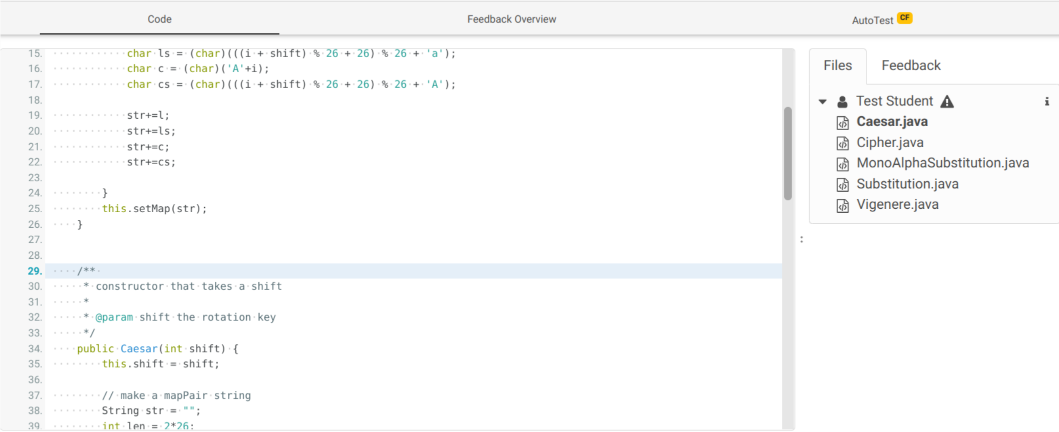
>	5	**Run Functionality Tests** Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionality`.	2 / 2	✓
>	6	**Run Full Functionality Tests** Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionalityFull`.	4 / 4	✓
	7	**Basic Commandline usage**	0.5 / 2	
>	7.1	**example run 1** Run `java Caesar encrypt 3 "The ships hung in the sky in much the same way that bricks don't."` and match its output to an expected value.	0.5 / 0.5	✓
⌄	7.2	**example run 2** Run `java Caesar decrypt 3 "Wkh vklsv kxqj lq wkh vnb lq pxfk wkh vdph zdb wkdw eulfnv grq'w."` and match its output to an expected value.	0 / 0.5	✗

Expected output ℹ

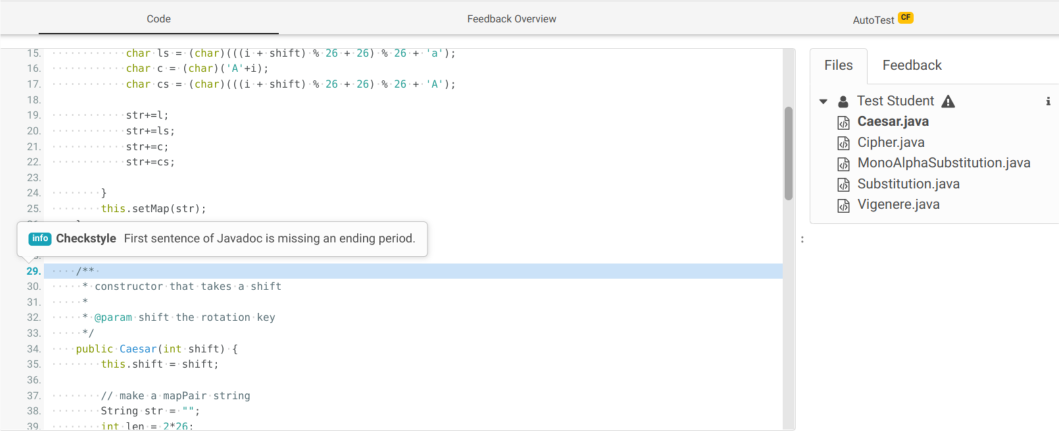
```
The ships hung in the sky in much the same way that bricks don't.
```

Actual output

```
The decoded message is: 
The ships hung in the sky in much the same way that bricks don't.
```



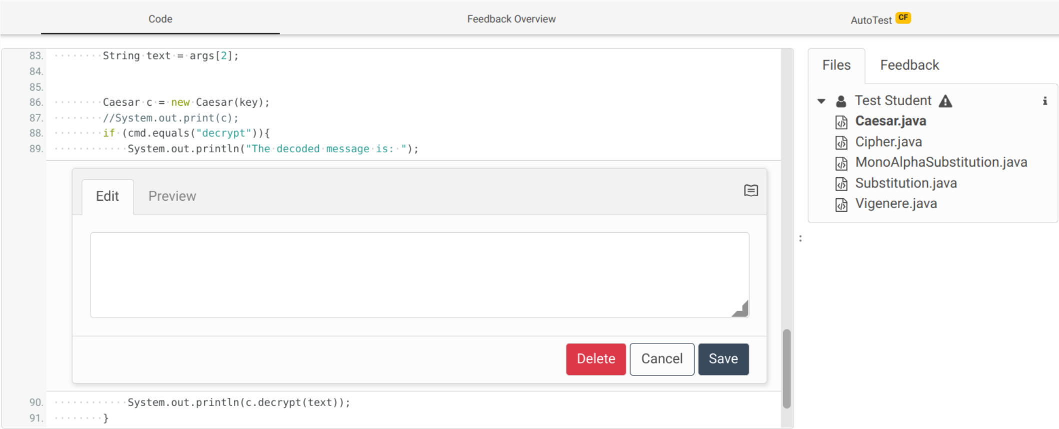
```
            char ls = (char)(((i + shift) % 26 + 26) % 26 + 'a');
            char c = (char)('A'+i);
            char cs = (char)(((i + shift) % 26 + 26) % 26 + 'A');

            str+=l;
            str+=ls;
            str+=c;
            str+=cs;

        }
        this.setMap(str);
    }


    /** 
     * constructor that takes a shift
     *
     * @param shift the rotation key
     */
    public Caesar(int shift) {
        this.shift = shift;

        // make a mapPair string
        String str = "";
        int len = 2*26;
```



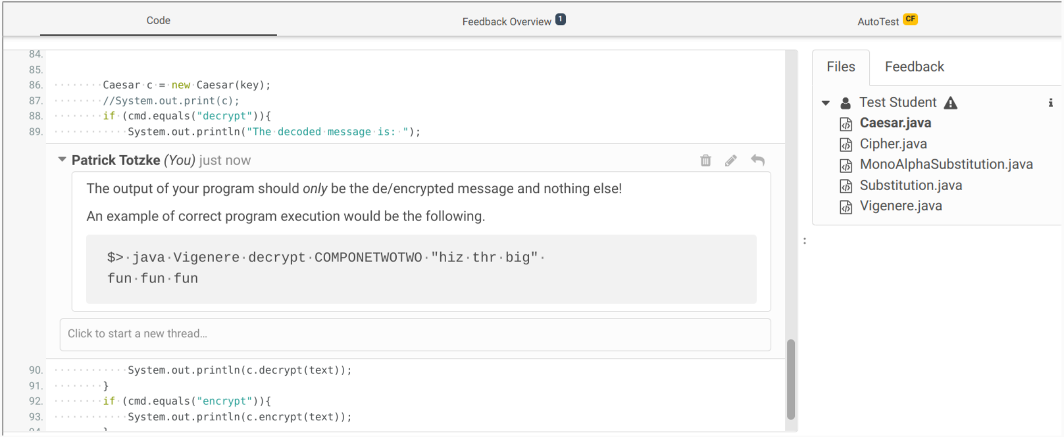
```
            char ls = (char)(((i + shift) % 26 + 26) % 26 + 'a');
            char c = (char)('A'+i);
            char cs = (char)(((i + shift) % 26 + 26) % 26 + 'A');

            str+=l;
            str+=ls;
            str+=c;
            str+=cs;

        }
        this.setMap(str);
```

info **Checkstyle** First sentence of Javadoc is missing an ending period.

```
    /** 
     * constructor that takes a shift
     *
     * @param shift the rotation key
     */
    public Caesar(int shift) {
        this.shift = shift;

        // make a mapPair string
        String str = "";
        int len = 2*26;
```



```
        String text = args[2];


        Caesar c = new Caesar(key);
        //System.out.print(c);
        if (cmd.equals("decrypt")){
            System.out.println("The decoded message is: ");
            System.out.println(c.decrypt(text));
        }
```



```


        Caesar c = new Caesar(key);
        //System.out.print(c);
        if (cmd.equals("decrypt")){
            System.out.println("The decoded message is: ");
```

Patrick Totzke *(You)* just now

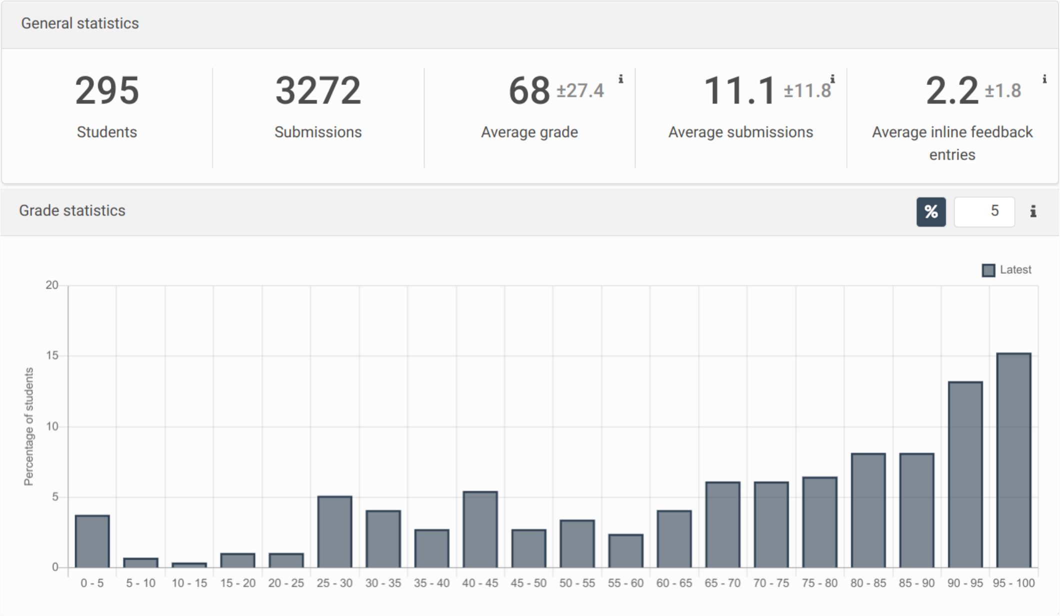
The output of your program should *only* be the de/encrypted message and nothing else!

An example of correct program execution would be the following.

```
$> java Vigenere decrypt COMPONETWOTWO "hiz thr big" 
fun fun fun
```

Click to start a new thread...

```
            System.out.println(c.decrypt(text));
        }
        if (cmd.equals("encrypt")){
            System.out.println(c.encrypt(text));
        }
```

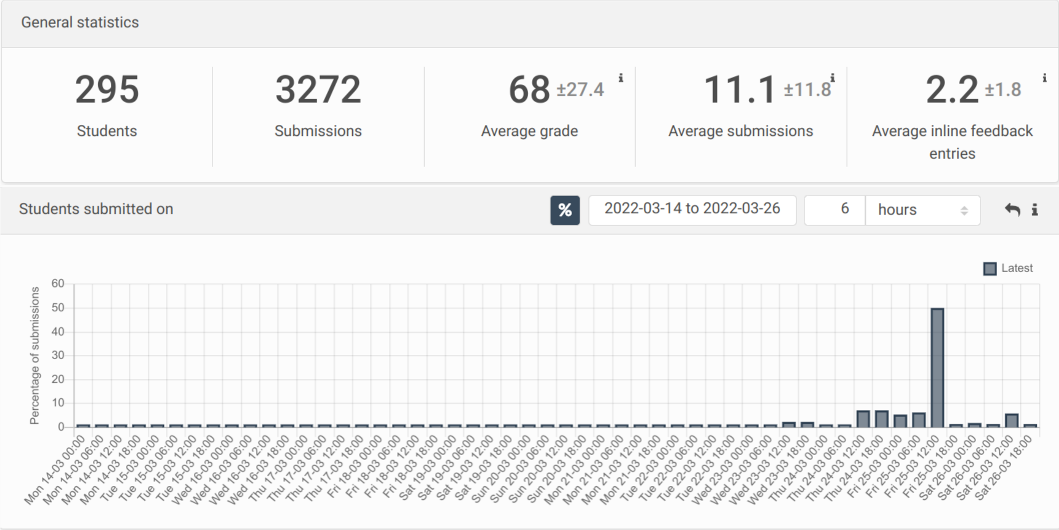


General statistics

295	3272	68 ±27.4 ℹ	11.1 ±11.8 ℹ	2.2 ±1.8 ℹ
Students	Submissions	Average grade	Average submissions	Average inline feedback entries

Grade statistics

% 5 ℹ



General statistics

295 Students

3272 Submissions

68 ±27.4 Average grade

11.1 ±11.8 Average submissions

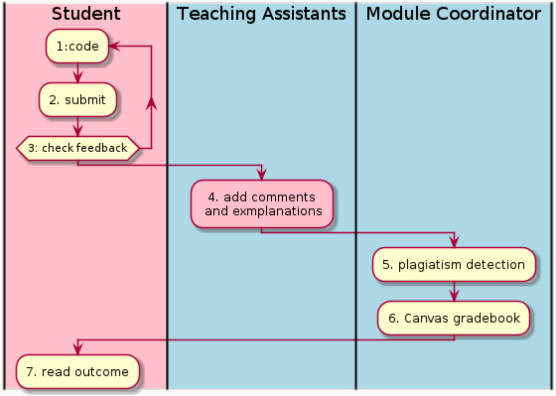
2.2 ±1.8 Average inline feedback entries

Students submitted on

% 2022-03-14 to 2022-03-26 6 hours

Impact

Assessment using CodeGrade



- **Quick!**
  Even for large modules.
- **Objective!**
  Checks and grading regime are the same for all students.
- **Consistent!**
  Students have a similar experience across modules;
- **lots of work**
  for module coordinator and markers but allows better use of their time.

Student Quotes

“CodeGrade is amazing, it not only helps us get instant feedback and identify areas we need to improve upon in real time but takes stress of us having to contact staff and beg for help instead of trying. CodeGrade promotes effort, it promotes actually finding a solution and engaging our brains which **makes coding fun instead of just learning**.”

Student Quotes

"Extremely helpful for assignments as I can test my code myself, but also with the given test cases (in case I missed one). It is definitely **something I would like to keep in the future**."

Student Quotes

“...the most impressive auto-testing system for any assignment I have ever had: Codegrade. What a brilliant creation! **It really made my life so much easier**. The commenting system was outstanding on it.”

Staff Quotes

“Automarking is an absolute must. Before CodeGrade even with significant help from TAs for marking, feedback was only released 2-3 months after the deadline. **Now it is almost instantaneous**.” (Rasmus Ibsen-Jensen, COMP207)

“CodeGrade has been a game changer for the assignments in my second-year module. **Student learning benefited significantly** from receiving continuous feedback from CodeGrade even before the assignment deadline and the marking time reduced significantly. My students just loved it. ” (Martin Gairing, COMP211)

Going Forward...

In Computer Science

All modules in the trial will continue to use CodeGrade.

Going Forward...

In Computer Science

All modules in the trial will continue to use CodeGrade.

In the wider Faculty

- Geo Sciences (ENVS363/563) could use it also CAs
- Materials ready for one EEE module (ELEC129 Intro to Programming in C)
- CodeGrade are happy to support further trials

Going Forward...

In Computer Science

All modules in the trial will continue to use CodeGrade.

In the wider Faculty

- Geo Sciences (ENVS363/563) could use it also CAs
- Materials ready for one EEE module (ELEC129 Intro to Programming in C)
- CodeGrade are happy to support further trials

Further dissemination?

Our findings are in prep for a publication in EdTech/Pedagogy journal plus a case study for the Centre for Innovation in Education (CIE).

Questions?

Questions?

Get in touch for more!

totzke@liverpool.ac.uk

Savings

module	students	assignments	mins saved/(a*s)	hours saved
COMP108	320	2	10	106.6
COMP122	369	3	16	295.0
COMP211	170	2	7.5	42.5
COMP226	200	2	12	80.0
COMP207	448	1	15	112.0
COMP282	104	2	15	52.0
COMP281	171	2	15	85.5
COMP517	150	3	30	225.0

Savings

module	students	assignments	mins saved/(a*s)	hours saved
COMP108	320	2	10	106.6
COMP122	369	3	16	295.0
COMP211	170	2	7.5	42.5
COMP226	200	2	12	80.0
COMP207	448	1	15	112.0
COMP282	104	2	15	52.0
COMP281	171	2	15	85.5
COMP517	150	3	30	225.0

$\approx$ **1000h**