Automated Feedback and Assessment of Programming Exercises

LT&SE Awards Showcase

Patrick Totzke
totzke@liverpool.ac.uk

Rahul Savani
rahul.savani@liverpool.ac.uk
What’s the Problem?
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!
Assessment of Programming Assignments

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

---

<table>
<thead>
<tr>
<th>Student</th>
<th>Module Coordinator</th>
<th>Markers(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. submit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. wait for all to submit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. download and distribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. wait</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. functionality checks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. grade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. consistency checks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. plagiarism checks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. email results</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. read feedback</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
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
- ✔ 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)
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

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

---

5
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?
### Setup

#### Categories

#### Part 1: Substitutions

<table>
<thead>
<tr>
<th>No</th>
<th>Summary</th>
<th>Score</th>
<th>Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Compile Substitution.java</strong> Run <code>javac Substitution.java</code> and check for successful completion.</td>
<td>0.5 / 0.5</td>
<td>✔️</td>
</tr>
<tr>
<td>2</td>
<td><strong>Only continue if compilation was successful</strong> Stop when you achieve less than 100% of the points possible.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>This verifies that you have uploaded a copy of the given <code>cipher.java</code> file and not edited it in any way. It is a prerequisite for everything that follows!</td>
<td>2.5 / 2.5</td>
<td>✔️</td>
</tr>
<tr>
<td>4</td>
<td><strong>Check signatures:</strong> 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!</td>
<td>12 / 12</td>
<td>✔️</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>Points</td>
<td>Score</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>5</td>
<td>Run Functionality Tests</td>
<td>Run the unit tests using <code>$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionality</code>.</td>
<td>2/2</td>
</tr>
<tr>
<td>6</td>
<td>Run Full Functionality Tests</td>
<td>Run the unit tests using <code>$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionalityFull</code>.</td>
<td>4/4</td>
</tr>
<tr>
<td>7</td>
<td>Basic Commandline usage</td>
<td></td>
<td>0.5/2</td>
</tr>
<tr>
<td>7.1</td>
<td>Example run 1</td>
<td>Run <code>java Caesar encrypt 3 &quot;The ships hung in the sky in much the same way that bricks don't.&quot;</code> and match its output to an expected value.</td>
<td>0.5/0.5</td>
</tr>
<tr>
<td>7.2</td>
<td>Example run 2</td>
<td>Run <code>java Caesar decrypt 3 &quot;Wkh vklsv kxqj lq wkh vnb lq pxfk wkh vdpv zdb wkdw eulfny grq'w.&quot;</code> and match its output to an expected value.</td>
<td>0/0.5</td>
</tr>
</tbody>
</table>
Run Functionality Tests Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionality`.

Run Full Functionality Tests Run the unit tests using `$FIXTURES/junit5.py run -- --select-class TestCaesarFunctionalityFull`.

Basic Commandline usage

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.

7.2 example run 2 Run `java Caesar decrypt 3 "Wkh vklsv kxqj lq wkh vnb lq pxfk wkh vdpw zdb wkdw eulfv grq'w."` and match its output to an expected value.

<table>
<thead>
<tr>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The ships hung in the sky in much the same way that bricks don't.</td>
<td>1. The decoded message is:</td>
</tr>
<tr>
<td></td>
<td>2. The ships hung in the sky in much the same way that bricks don't.</td>
</tr>
</tbody>
</table>
```java
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);
```
char ls = (char)(((i + shift) % 26 + 26) % 26 + 'a');
char lc = (char)('A'+i);
char cs = (char)(((i + shift) % 26 + 26) % 26 + 'A');

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

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;
```java
83. String text = args[2];
84.
85.
86. Caesar c = new Caesar(key);
87. System.out.print(c);
88. if (cmd.equals("decrypt")){
89. System.out.println("The decoded message is: ");
90. c.decrypt(text);
91. }
```
Caesar c = new Caesar(key);
//System.out.print(c);
if (cmd.equals("decrypt")){
    System.out.println("The decoded message is: ");
}

Patrick Totske (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
$>java Vigenere.decrypt.COMPONETWO.TWO."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

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>295</td>
</tr>
<tr>
<td>Submissions</td>
<td>3272</td>
</tr>
<tr>
<td>Average grade</td>
<td>68 ±27.4</td>
</tr>
<tr>
<td>Average submissions</td>
<td>11.1 ±11.8</td>
</tr>
<tr>
<td>Average inline feedback entries</td>
<td>2.2 ±1.8</td>
</tr>
</tbody>
</table>

### Students submitted on

<table>
<thead>
<tr>
<th>%</th>
<th>2022-03-14 to 2022-03-26</th>
<th>6 hours</th>
</tr>
</thead>
</table>

### Percentage of submissions over time

[Graph showing submission percentage over time]
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.
“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.”
“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.”
“...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.”
“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)
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

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

\[\approx 1000h\]