GLASGOW CALEDONIAN UNIVERSITY • FOR THE COMMON WEAL •

GCU Learning and Student Experience Showcase, 24/1/18

Developing online practical activities and automated assessment for programming with Codio

Jim Paterson (SEBE)

GCU
Glasgow Caledonian
University

University for the Common Good

Practical assessments in programming

- Practical assignments involving programming are common in Computing and in an increasing range of other programmes
- Marks are typically awarded, at least partially, on the basis of the correct functioning of the program
- Testing student programs can be time-consuming, particularly when classes are large
- As a result it is difficult to give timely feedback to students

Automated marking

- Automated marking involves the functioning of student’s program being tested by another program which then calculates a correctness score
- Can often be done by writing test scripts and applying these to the student’s code rather than running the code manually and observing the output
- There are some purpose-built systems and learning environments that integrate programming activities and marking of these
- These are not particularly widely used, systems often developed “in-house” or targeted at specific topics

Motivations for using automated marking now

- Increased number of students
- Increased importance of online learning, e.g. for off-campus/transnational students
- System selected for evaluation, **Codio**, offers wide range of capabilities that are potentially useful:
    - Integrates programming activities, online content and learning management in browser-based application
    - Allows a range of assessment types, including programming assignments, to be set
    - Allows instructor to see student’s work at any point “live” for support or assessment
    - Simplifies setting up programming environments for a wide range of scenarios

Automated marking – expected benefits

- Complete uniformity in marking
- Immediate feedback during task and after submission
- Students can correct (and learn from!) their mistakes before mark is final
- Students know what mark to expect when they submit their work
- Fewer complaints about marks
- Students learn importance of testing against requirements
- Reduced effort in marking

Automated marking – limitations/possible pitfalls

- Only assesses functioning of program, may want to manually mark other aspects (e.g. coding style, documentation of code)
- Students may try to “game” the tests (program to just pass the tests rather than fully solve the problem assigned
- Can promote plagiarism
- Initial effort in creating assessments can be significant

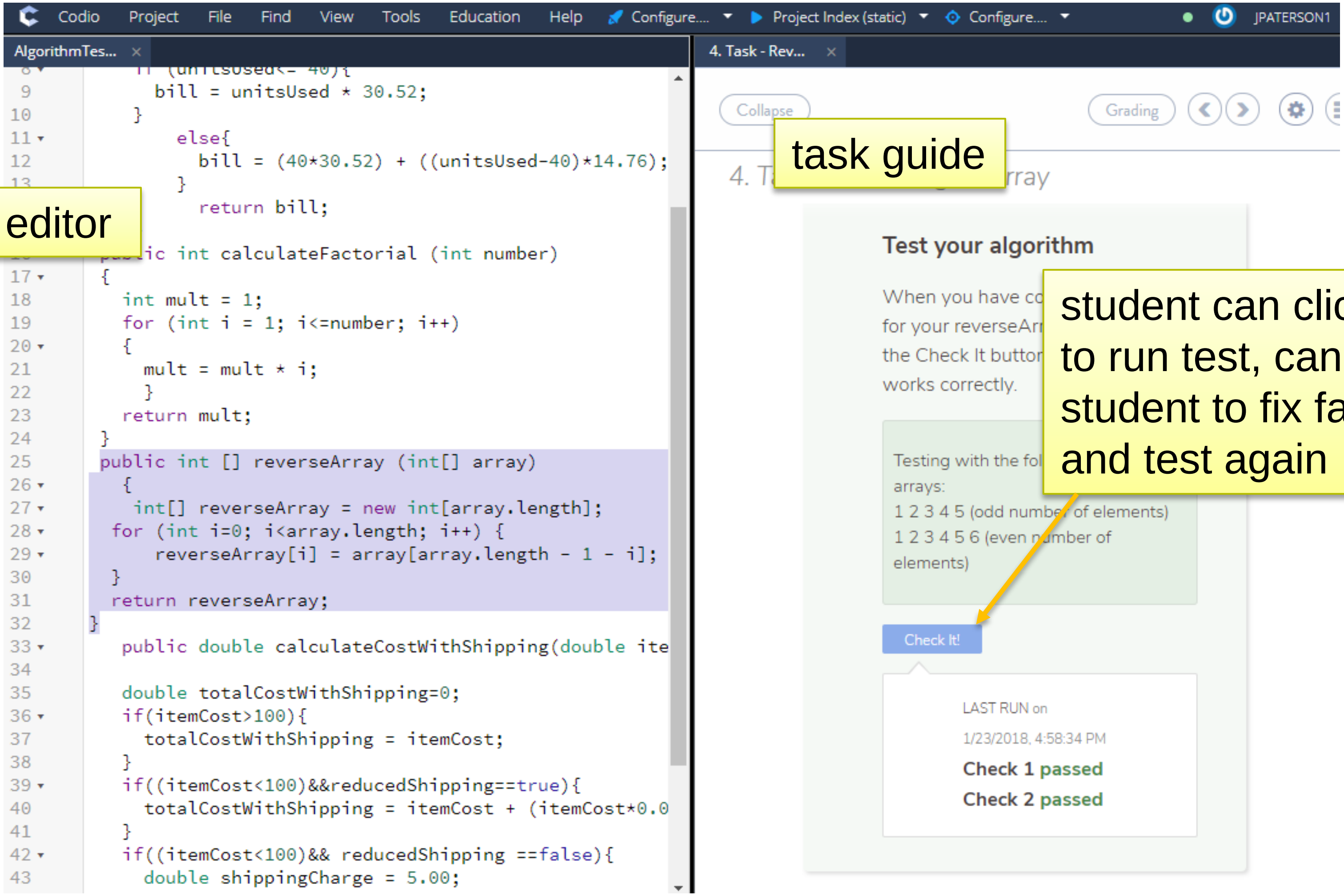
Progress

- Trialled with one module at ALC in Tri A 17/18
- Assessed lab activities a mix of "traditional" and Codio-based exercises
- Outcomes:
  - Students received immediate feedback on Codio assessments
  - Students enjoyed using Codio – communicated this through facilitators, SSCG and MEQ, e.g. from MEQ:

    "the use of codio was a good experience that enabled step by step learning"
  - Learning curve and time to translate from traditional task specs to Codio exercise with auto-marking was significant

GCU Glasgow Caledonian University

Example of an assessment task in Codio



```
    if (unitsUsed<= 40){
      bill = unitsUsed * 30.52;
    }
       else{
         bill = (40*30.52) + ((unitsUsed-40)*14.76);
       }
         return bill;

public int calculateFactorial (int number)
{
  int mult = 1;
  for (int i = 1; i<=number; i++)
  {
    mult = mult * i;
    }
  return mult;
}
public int [] reverseArray (int[] array)
  {
   int[] reverseArray = new int[array.length];
 for (int i=0; i<array.length; i++) {
     reverseArray[i] = array[array.length - 1 - i];
 }
 return reverseArray;
}
  public double calculateCostWithShipping(double ite

  double totalCostWithShipping=0;
  if(itemCost>100){
    totalCostWithShipping = itemCost;
  }
  if((itemCost<100)&&reducedShipping==true){
    totalCostWithShipping = itemCost + (itemCost*0.0
  }
  if((itemCost<100)&& reducedShipping ==false){
    double shippingCharge = 5.00;
```

4. T rray

Test your algorithm

When you have co for your reverseAr the Check It buttor works correctly.

Testing with the fol arrays:
1 2 3 4 5 (odd number of elements)
1 2 3 4 5 6 (even number of elements)

LAST RUN on
1/23/2018, 4:58:34 PM
Check 1 passed
Check 2 passed

code editor

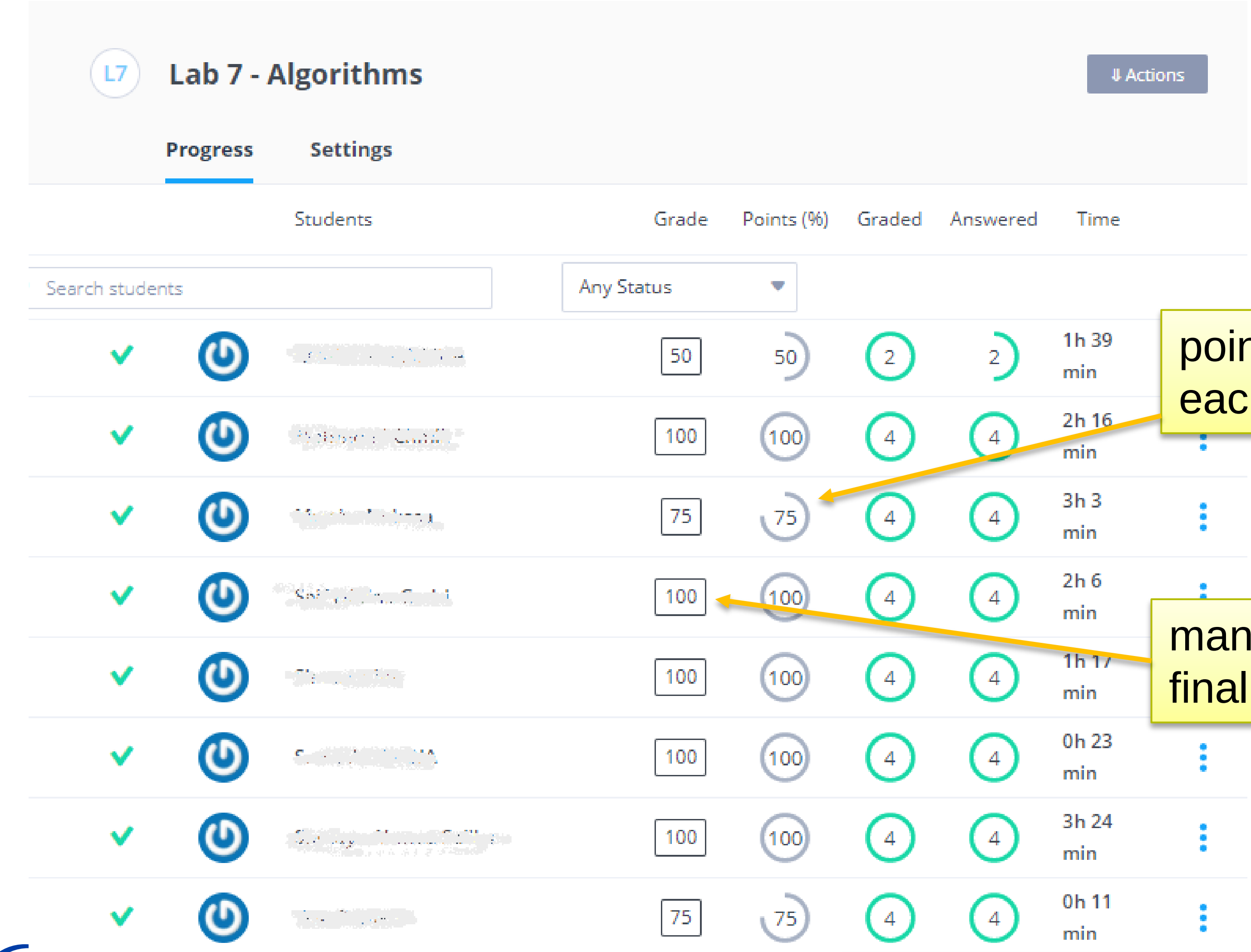
task guide

student can click *Check it* to run test, can allow student to fix failed test and test again

screenshot shows instructor view of a specific student's work, after successful completion in this case but can view at any time

GCU Glasgow Caledonian University

Assessment results recorded in Codio



	Students	Grade	Points (%)	Graded	Answered	Time
✓	[redacted]	50	50	2	2	1h 39 min
✓	[redacted]	100	100	4	4	2h 16 min
✓	[redacted]	75	75	4	4	3h 3 min
✓	[redacted]	100	100	4	4	2h 6 min
✓	[redacted]	100	100	4	4	1h 17 min
✓	[redacted]	100	100	4	4	0h 23 min
✓	[redacted]	100	100	4	4	3h 24 min
✓	[redacted]	75	75	4	4	0h 11 min

points recorded for each test passed

manual override for final grade possible

Marks can be released to GCU Learn grade centre

GCU
Glasgow Caledonian
University

Conclusions

- Creating excellent online learning experiences requires the use of appropriate tools, e.g. for content delivery, assessment and feedback
- Codio promises to be a valuable tool specifically for learning programming and related topics
- Students have enjoyed the initial experience with it
- Need to invest in appropriate tools and the time to develop expertise in their use

Thank you.

Glasgow Caledonian University

University for the Common Good