University of Aberdeen

School of Natural and Computing Sciences

Department of Computing Science

BSc in Computing Science

2023 – 2024

Assessment 2 – Student Budgeting Tool

Title: CS2020 – Software Programming	Note: This assessment accounts for 35% of your total mark of the course.

Learning Outcomes

On successful completion of this component a student will have demonstrated competence in the following aspects:

- Ability to write and run basic Java applications
- Ability to judge how Java applications should be structured
- Understanding and ability to apply, techniques to support correct code in a Java application
- The ability to analyse simple problems and design a program solution
- The understanding of software libraries and their application in order to classify them for usage
- The ability to apply programming concepts in order to support unit testing with JUnit
- Knowledge and understanding of basic programming concepts and their application
- Ability to create a graphical user interface using Swing

A note on the code, and your approach to solve these challenges.

Swing applications do not always return values. You will see some methods in the starter code, which are void methods, and do not return values. This means they cannot easily be tested by JUnit. Furthermore, this makes it harder to separate the functionality of the application from its GUI. Take care to separate the logic/algorithms of the application from the GUI. This will make it easier for you to write the application.

Information for Plagiarism: Your report and test cases may be submitted for plagiarism check (e.g., Turnitin). Please refer to the slides available at MyAberdeen for more information about avoiding plagiarism before you start working on the assessment. Please also read the following information provided by the university: https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/

Use Git to Save Your Work

While we're not marking your use of Git in this assignment, you are strongly encouraged to do this work inside a Git repository. By doing that you gain the benefit of rolling back your edits, or doing work in branches, and then merging it to the main branch as you accomplish tasks along the way.

Perhaps most importantly, by using Git and pushing your work to a remote repository on GitHub, you provide a safe backup of your work. You'd be surprised by how regularly fellow students discover failed hard disk drives, or that the laptop they're using needs to be returned to its owner, or be repaired due to some liquid being spilt on the keyboard.

If you push your work regularly to GitHub, then you can easily (a) pull it into Codio and carry on if you lose your laptop, and (b) move it to a different device if you need to.

By the way, 'regularly' in this case means making a commit every 5 or 10 minutes locally, and then pushing it to GitHub every hour. Small local commits mean you can roll back small changes without much trouble, and pushing remotely less frequently stores your work safely offsite.

Assessment Details

In this assessment, you will build a simple budgeting tool for students, which is displayed in a graphical user interface. We will provide a simple starter system which displays a UI with two input text boxes, which are calculated for total income and then displayed in output text box for income when a Calculate button is pressed. Using the starter is optional, you can start from scratch if you prefer. In any case you must ensure that you cover these same components.

Functionality and marking are as described below.

Basic system (30 pts)

- 8 pts: user enters three (or more) income fields (wages, loan, sales, other, etc) and the system computes total income when a Calculate button is pressed.
- 7 pts: user enters three (or more) spending fields (food, rent, commuting, other, etc) and the system computes total income when a Calculate button is pressed.
- 5 pts: When Calculate is pressed, the system also shows surplus/deficit (income minus spending). This is black if positive or zero, and red if negative.
- 5 pts: System checks user input (numbers for validity), and produces an appropriate error message if input is not valid. Empty fields are treated as 0 (no error message)
- 5 pts: A good collection of JUnit tests to ensure methods return correct values.

Extensions (20 pts)

- 8 pts: allow users to specify income/expenditure numbers per week, per month, or per year, for each input field. This requires both adding appropriate choice widgets (such as combo boxes), and also modifying the way you calculate totals. You can assume that there are 52 weeks in a year, 12 months in a year, and 4.3333333 weeks in a month.
- 8 pts: implement "spreadsheet" behaviour, that is totals are updated whenever the user changes a number or time-period, with no need to press a Calculate button. You should update whenever the focus shifts, as well as whenever an action is performed.
- 4 pts: A suitable collection of JUnit tests for the methods supporting these actions.

Undo (30 pts)

- 10 pts: implement a single-level of Undo, so the user can "undo" his or her most recent action. This require saving the state of the system (ie, all numbers) when a change is made.

- 10 pts: implement multiple Undo, so the user can undo multiple actions. We recommend that you create a class to hold state information numbers, and then maintain a stack of states.
- 10 pts: Good collection of JUnit tests for Undo

Programming style (20 pts)

- 10 pts: Java code follows good coding style, including good code comments and variable names
- 10 pts: Java code is well structured and decomposed into appropriate methods, classes, and other structures that you might use.

Submission Instructions

This assessment is due at **5PM** on **Thursday, 7 December 2023**

Please submit in Codio

- Coversheet for the assessment
- Your code, which must run as a maven project
- A README file explaining what you have implemented, and challenges you encountered (text or PDF)

It is your responsibility to ensure that your code runs in Codio. If you do it on another device, and then move it to Codio, you need to ensure it runs there, and that it runs as expected.