

Java Programming Revision Tool

Nicole Kearns

March 11, 2014

Contents

1	Requirements	2
1.1	Student Surveys	2
1.1.1	Level 4 Student Survey	2
1.1.2	Level 2 Student Survey	3
1.2	Functional Requirements	5
1.2.1	Requirements Table	5

1 Requirements

This section discusses the requirements elicitation process for our project. As I have an client, this process is important to ensure that all requirements are identified in order to produce the application that fully meets the clients needs.

1.1 Student Surveys

To better my understanding of what would be the most effective tools and features for learning and revising Java programming, I asked a sample of Level 2 and Level 4 students to carry out a survey. The summarised results from both surveys are discussed below.

1.1.1 Level 4 Student Survey

In order to gain a better understanding of what students thought of the Java Programming 2 course and to see what features would be beneficial for students learning Java, I spoke to a sample of 10 Level 4 Computing Science students to find out what they thought about the course when they took it 2 years previously. I asked a group of 10 students the following 3 questions:

1. How useful did you find the material provided for JP2?
2. What other tools did you use to complete the lab exercises or to learn Java, if any?
3. If you could do the course again, what would you do different? What other tools would you like to be available?

Having recorded the answers from the survey, I have analysed them and have arrived at the following conclusions:

From the answers given to question 1 it would seem that, overall, the students thought the material was quite useful, however some improvements could be made. Some students thought that the lecture notes were useful for introducing the basic concepts and syntax of Java. However, when the material became more difficult, the lecture notes lacked detailed information and examples to show how they are used.

Most students found that the lab exercises were more beneficial than the lectures as they were able to practice writing Java code, instead of merely reading it. All participants, with one exception, also used extra resources when completing the lab exercises. An example of such is if they came across a problem they were unable to solve. The following resources were the more common ones which were discussed; general internet searching, StackOverflow, and online tutorial videos.

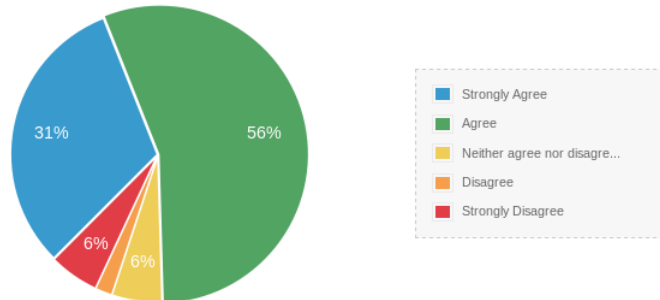
When question 3 asked what they would have liked to have been made available, most participants suggested that it would be useful for more examples or snippets of code to be provided within the lectures. This would allow students to understand more fully how the concepts are used in practice. From the feedback given during the survey, consideration could be given to make the lectures more interactive to help the students to stay interested and allow them to practice and improve hand-writing code. Another suggestion was to provide the students with short video tutorials to talk students through more complicated concepts and provide a step-by-step walk-through of how it works and how it is used.

Overall, it would seem that the students thought the lectures were adequate, they provided enough information to be useful for exam revision, however they didn't provide enough examples to demonstrate how the concepts are used. From the results collected, I believe that students find interactive tools more useful when learning a programming language as it allows them to practice and improve on writing correct Java code.

1.1.2 Level 2 Student Survey

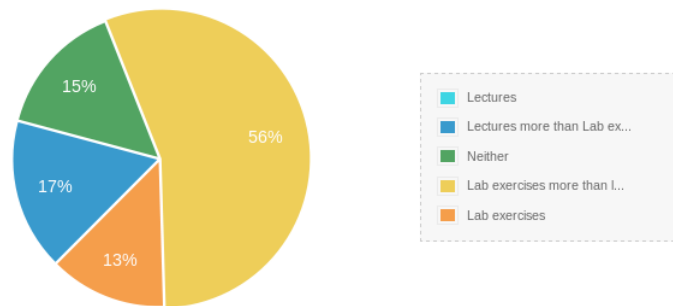
To further my understanding of what the second year students think about the current teaching methods of Java within the Java Programming 2 (JP2) course, I asked them to fill out a survey with the following questions:

1. The material provided for the JP2 course is useful. (On a scale of strongly agree to strongly disagree)
2. What do you find more useful, the lectures or the lab exercises?
3. What other resources have you used to understand Java concepts and syntax?
4. What resources do you use when you have a specific problem?
5. Is there any other resources you would like to be available within this course to help you learn and revise Java?



Question 1 Results

From the graph above, you can clearly see that majority of students think that the material provided for the course is very useful, with very few people thinking that it is not.

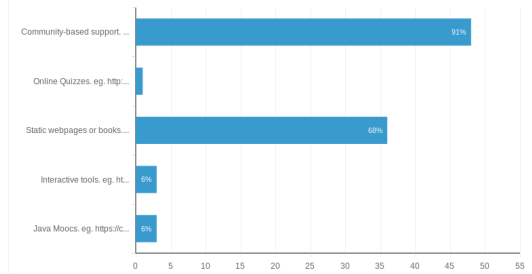
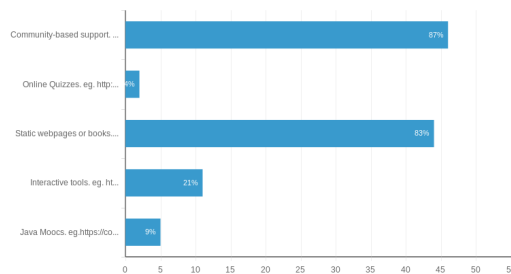


Question 2 Results

For question 2, when asked what they found more useful, lecture notes or lab exercises, it can be said that a vast majority of students find that the lab exercises are more useful, assuming that this is because this allows them to practice writing code.

For questions 3 and 4, it would seem that students use the same resources for both learning Java and for referring to when they encounter a problem. The resources which are most popular with the students are community-based support and static webpages/books. Community-based support allows the user to ask questions or post problems that they are struggling with, answer questions from other people, and simply view problems and solutions from other users.

A few suggestions were given for tools and facilities which they would like to be available are:



- Video tutorials.
- Mini exercises to create fun mini projects.
- The code provided in class to be provided in the notes or on moodle.
- Plenty of exercises and examples, especially to emphasise OO programming.
- A more simplified explanation of the basics.

1.2 Functional Requirements

From the information and suggestions provided by both the Level 2 and Level 4 Computing Science students, my client and I came up with the following activities which we think would be most beneficial to the students when learning and revising Java programming:

- **Multiple Choice Questions** in which the user is provided with questions about Java syntax and snippets of code in which they must predict the output. This allows the user to improve their ability to read lines of code and understand it's behaviour.
- **Fill in the Blanks Questions** which provide the user with a small sample of code which a Java keyword removed and the user must enter the keyword which will complete the line of code.
- **Find the Errors.** This provides students with a piece of code, which contains errors and the user must identify where in the code the errors lie. This will enhance the users ability to read code, predict it behaviour and identify where the errors occur.
- **Code your Own**, a section which provides the user with a question or scenario in which the user must write the own code in order to complete the task.
- **Video Tutorials**, to provide the students with guided help of the basic Java concepts and syntax.

- **Tutorials**, to allow the students to read over the basic concepts and syntax for Java programming.

1.2.1 Requirements Table

Student - Uses the application to learn and revise Java using the activities provided; answering the multiple choice questions, completing the Fill in the Blank questions, identifying the errors within the broken code provided and writing their own code. Also through browsing the tutorials and video tutorials which will go over the basic Java syntax and concepts.

Lecturer - Responsible for adding content to the application such as the questions for all relevant activities, the tutorials and video tutorials.

Student Requirements

Id	Requirement	Description	Priority
SR1	View Tutorials	The user must be able to view the tutorials containing information about the basic Java concepts and syntax.	Must Have
SR2	View Video Tutorials	The user must be able to view the video containing information about the basic Java concepts and syntax.	Must Have
SR3	View Questions	The user must be able to view the different questions within Multiple Choice, Fill in the Blanks, Code Your Own and Find the Errors.	Must Have
SR4	Answer Questions for Multiple Choice and Fill in the Blanks	The user must be able to answer the different questions within Multiple Choice, Fill in the Blanks, Code Your Own and Find the Errors.	Must Have
SR5	Write Java code and execute it for Code Your Own	The user must be able to write Java code and execute it, receiving the results within the application as to whether or not it was successful.	Must Have
SR6	Edit the provided Java code and execute it for Find the Errors	The user must be able to edit Java code provided to correct the errors and then execute the code, receiving the results within the application as to whether or not it was successfully compiled.	Must Have

Lecturer Requirements

Id	Requirement	Description	Priority
LR1	Add Tutorials	The Lecturer must be able to add new tutorials to the application.	Must Have
LR2	Edit Tutorials	The lecturer should be able to edit the existing tutorials within the application.	Should Have
LR3	Delete Tutorials	The lecturer must be able to delete current tutorials from the application.	Must Have
LR4	Add Video Tutorials	The Lecturer must be able to add new video tutorials to the application.	Must Have
LR5	Edit Video Tutorials	The Lecturer must be able to edit existing video tutorials within the application.	Should Have
LR6	Delete Video Tutorials	The Lecturer must be able to delete existing tutorials from the application.	Must Have
LR7	Add Questions	The Lecturer must be able to add new questions for all relevant activities to the application.	Must Have
LR8	Edit Questions	The lecturer should be able to edit questions for all relevant activities from within the application.	Should Have
LR9	Delete Questions	The lecturer must be able to delete questions from the application.	Must Have