

studysieve *student guide*

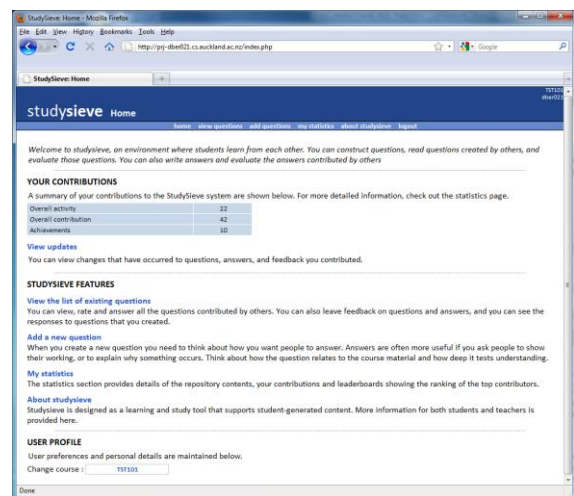
studysieve contribution

studysieve is designed to help students learn and revise course material. The resource material (questions and answers) that you create will be shared with your classmates and their questions and answers will be available for you to use in your own study.

studysieve is focused around exam-style questions. This makes it an excellent tool for studying material leading up to the final exam. Developing questions and answers that involve code which is not compiled and executed provides essential practice with explaining code, and generating solutions in an exam-style setting.

Home page

After logging in, you will be shown the home page for studysieve. The main menu allows you to view questions that have already been submitted, add a new question, view a summary of your own contributions, and find out more information. Although you have to log in to use the system, all contributions are confidential (i.e. none of the other users can see who has contributed a particular question, answer, rating or feedback).

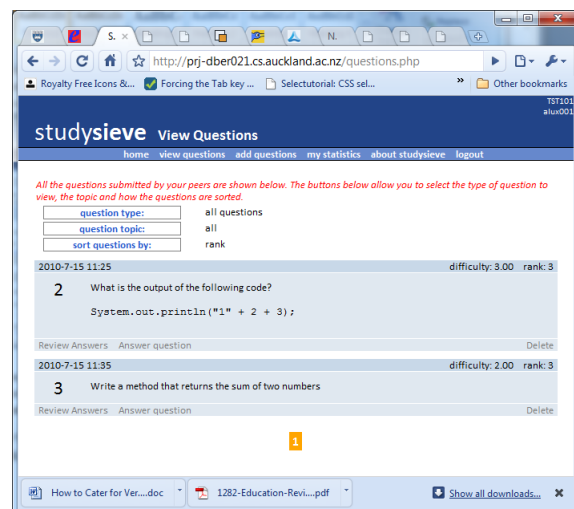


Viewing the list of existing questions

By default, all of the questions submitted to studysieve are displayed in a single list. You can choose to limit the list, so it only shows you questions that you have already answered, questions that you have not answered yet, or questions that you have created yourself. You can further filter the list so that it only contains questions about a particular topic (you can select from a list of topics, or use a search function that searches the question text).

You can also decide how to sort the list. You can order the questions by the number of answers submitted to the question, the rank of the question, the date it was created, or the level of difficulty.

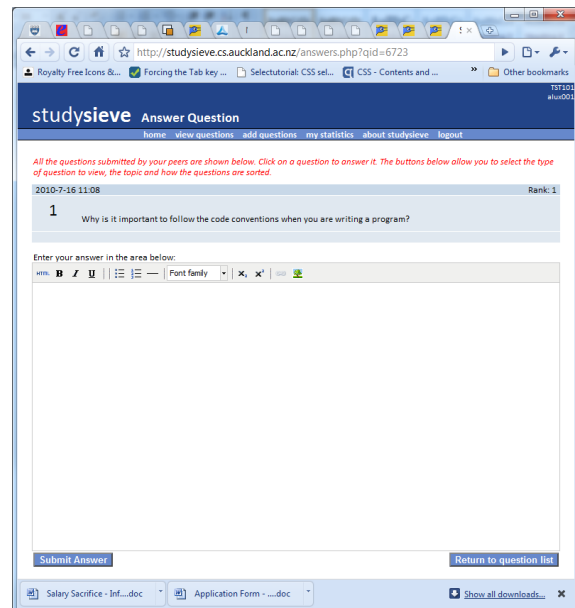
On the left side is a large number showing the number of answers submitted to the question. At the top of each question is the date the question was created, the difficulty rating, and the rank of the question (based on the ratings of other students about how helpful the question is for learning and revision).



Answering a question

When you are ready to answer a question, start the process by clicking on the link to answer the question (located directly below each question). You will be directed to a page that allows you to answer the question.

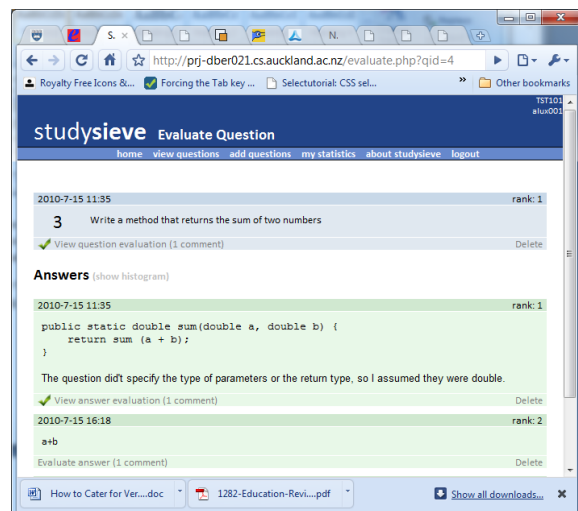
Note that you can use simple formatting tools to help make your answer clearer. If you are writing computer code, then make use of the monospace font. If you have something to highlight then you can use bold or italics to highlight the text. You can also insert a link to an image on the web (you can't upload a new image directly to studysieve, but you can link to an image hosted elsewhere).



Viewing other answers

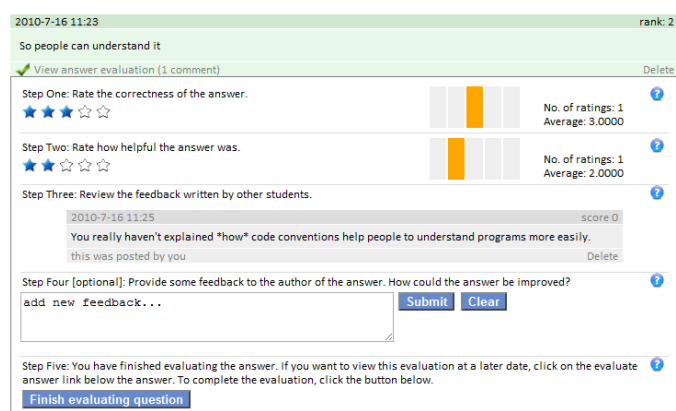
Once you have submitted your own answer, the answers of other students will be displayed.

Look carefully and critically at the answers submitted by other students. Try to determine which of the answers are better than others, and what exactly makes them better answers. It is important that an answer is correct, but some correct answers have a clearer explanation, show their working, or just have better style. Which of the answers show you clearly that the author understands the material?



Evaluating the question and other answers

You are required to evaluate the question and a sample of the other answers before you can continue. When you evaluate the question or answer, you will be required to rate it on a 5 point scale. Once you rate something, you will be able to see how others have also rated it. Perhaps the most important part of the evaluation process is the feedback provided through the open-ended comments. Try to give some useful feedback to the author (and to other students who are reading the evaluations) about the quality of the answer.



Adding a question

After you have answered a few questions, it is time to add your own question. When you create a question you first need to write the question text. The system will search the repository for the text you have entered and will produce a list of similar questions that already exist. Once you have entered your question, check to make sure a similar question is not already present (if the same question already exists, then try a different question).

Once you have a good question, write the best sample solution you can, including a good explanation of the answer.

Finally, add a set of topics related to the question that will help other students find your question. You can select from the existing topics by clicking on them, or add your own new topics.

Before you add a question, review the learning outcomes of the course and try to think how you would ask a question that assesses whether a student has

achieved the outcome or not. There are many different kinds of questions you could ask. Try to write a relevant, but interesting question, and make sure your answer to the question is not only correct, but helps other students to learn! A brief discussion of question types is provided below.

Question

Enter your question in the area below. When creating your question, consider the following:

- Is your question clear and easy to understand?
- Have you explained exactly what you want the reader to do?
- Have you checked the spelling and grammar?
- Are you using correct indentation and monospace font for code?

HTML **B** *I* U | | | | | | | | | | Font family | x₂ x² | | | |

[show similar questions :](#)

Answer

Enter your answer in the area below. When creating your answer, consider the following:

- Have you answered the question correctly?
- Will your answer help a reader to learn more about the topic?
- Have you checked the spelling and grammar?
- Is there any way that your answer could be improved?

HTML **B** *I* U | | | | | | | | | | Font family | x₂ x² | | | |

Topics

Select the topics that relate to this question. You may select multiple topics.

code convention(1) java(2) test(1)

You may add additional topics in the field below. To enter multiple topics, separate each with a comma.

Preview

When you are ready to share your question, click on the preview question button below.

The simplest kind of question requires memorization of facts. These kinds of questions should be used sparingly, since they can be useful for revision, but tend not to improve understanding.

Some typical examples are:

- List the arithmetic operators in increasing order of precedence
- Define the purpose of a constructor.
- Describe the difference between a while and for loop.

Questions that test understanding frequently ask students to explain something in their own words, or to translate from one form to another. Some typical examples are:

- Explain in plain english what the following code does.
- Convert the following algorithm into Java code.

Many questions assess the ability to carry out a process. These questions often begin with an instruction to perform the relevant process. Some examples are:

- Evaluate the following expression.
- State the output of the following code.

Questions that probe deep understanding often ask students to analyse or evaluate a concept introduced in the course. For example:

- Which of the following methods is better? Explain why.

The questions that involve the highest levels of ability ask students to create a solution to a new problem they haven't seen before. For example:

- Write a method that accepts a String as a parameter and returns the String with all the vowels removed.

When you write material to add to studysieve, remember that the material you write will be used by your peers for learning and revision. You should aim to produce professional, high quality material that will make a real contribution to your learning community.

<http://studysieve.cs.auckland.ac.nz>