

# CITS1001 project 1 2014

*This page is a copy of the README.txt file in the project package.*

## Purpose of the project

The project is intended to extend your understanding of object collections, and give you experience in completing and demonstrating a complete Java application. The project can be tackled after completing the CITS1001 lectures and labs for weeks 1 to 5, covering material up to Section 4.6 of Chapter 4 of Objects First with Java - A Practical Introduction using BlueJ

Fifth edition

David J. Barnes and Michael Kolling

Pearson Education, 2012

## The task

A collage is a piece of art made by sticking various different materials such as photographs and pieces of paper or fabric onto a backing. In this project you will write Java code for making collages out of images stored on a computer.

Your task for this project is to complete the outline implementation of the Collage class, and to implement a new class, MyDemo, to illustrate the functionality of your project. Optionally, students who have extra time and wish to extend their programming skills are encouraged to identify and implement some challenge tasks for the project.

The project application comprises 6 classes: Fragment, Collage, ImageManager, CollageTest, Canvas and SampleDemo. A collage is modelled in Java as a collection (ArrayList) of image Fragment objects. A complete version of all the necessary classes (except Collage and MyDemo) are already provided for you in this project, and they should not need any modification.

Your first task is to complete the Collage class. An outline (walking skeleton) of this class has been provided in this BlueJ project. Implement code for each method using information given in the Javadoc comments of the source file, and confirmed by the JUnit test class, CollageTest, that is also provided.

Your second task is to develop your own MyDemo class to illustrate the functionality of your project. This class is intended to be shown to a customer. It should illustrate that each of your Collage methods works correctly. The provided SampleDemo class gives some ideas about how you could do this.

**SELF CHECK:** Your project will be marked by running the CollageTest, Canvas, and SampleDemo classes that were distributed with the project. You must ensure that your submitted code runs with these three marking classes. If you complete the challenge section, then you may make changes to the Fragment and ImageManager. However, you must not change the method signatures of the published versions. You can change the implementation of given methods or you can add new methods of your own. These restrictions are imposed because object-oriented applications in Java rely on interactions between existing and new classes. Therefore, learning how to work with given class interfaces is an important part of learning to use Java effectively.

## Possible Staged Implementation

The project has been broken down into suggested separate stages to help you create the finished version in small steps. You are recommended to compile, test and run the application after each stage to check

that the changes you have made are correct. You can gain good marks even if you do not complete all the methods, so long as the code you have written compiles and runs correctly. But if you submit a large body of code that does not compile or that crashes, then few marks can be awarded.

1. **Read the code and its documentation for each class in the BlueJ project** to familiarise yourself with the methods offered. When you open each class, toggle between the Source Code and Documentation views (the top right hand of the window) to see a summary of the services offered by each class.
2. **Implement the required methods in the Collage class one by one.** Test your code after each new method is added using the Junit tests in CollageTest and by running the SampleDemo provided.
3. **Create your own MyDemo class** and write code so that running your demonstration clearly illustrates the capabilities of your Collage class. The provided SampleDemo class can be used for ideas. Use your imagination here to make an interesting collage. You will NOT be marked on your artistic flair, but you should ensure that your demonstration does illustrate the functionality that you have implemented. You can use the images from the images directory provided with the project, or source your own, for example, from [google images](#).
4. **CHALLENGE TASKS (optional)** Completing steps 1, 2 and 3 (above) is worth 80% of the total marks for the project. If you have time and wish to extend your knowledge of Java, then you are encouraged to extend your project to take it further towards a complete application. Up to 20% will be awarded for the completion and demonstration of challenge tasks. There are only a few project marks to be awarded for the Challenge tasks (relative to the effort required): this part of the project is primarily intended for interest and fun.

It is up to you to identify, select and implement challenge tasks. Submit a TEXT file, Challenge.txt, giving a brief summary of your challenge code: what you have implemented and how it works. Some possible additions could be new ways of manipulating collages or additional functions for manipulating images. You may add new classes to the project, or add new methods to given classes, but you must NOT change the signatures for any of given methods in the original project. As well as submitting the written summary, ensure that your MyDemo class shows the capabilities of your challenge code.

If you are completing the challenge tasks, then to learn more about 2D graphics in Java try the [Java tutorials](#) For using colours in Java see the colorinfo guide on the Resources section of the unit web pages (and included in the project directory) or search for online tutorials or see the text book.

## Submission

Submit the following files to csubmit by the PROJECT DEADLINE given in csubmit. Note that you must submit your code at the end of the week before the demonstrations. It is allowed to make further changes to your code before the demonstration, but your submitted code will be marked for the other components.

1. Collage.java
2. MyDemo.java

If you have completed any challenge tasks, then also submit

3. The .java files for any additional classes you have implemented or classes you have modified
4. A TEXT file, Challenge.txt, giving a brief summary of your challenge code: what you have implemented and how it works.

**Remember that the published versions** of Canvas, SampleDemo or CollageTest will be used to test

your project.

Marks will be deducted for project files submitted after the deadline, or if files are submitted incorrectly and a resubmission is needed. 35 out of 50 marks will be awarded for the code submitted to csubmit by the deadline.

If you wish you may continue working on your code after it is submitted up to your demonstration. 15 out of 50 marks are awarded for the demonstration.

In accordance with the departmental policy on plagiarism, all submitted lab work is expected to be the sole effort of the student submitting the work. All submissions will be checked for similarity and the consequences for misconduct can be severe, including being awarded 0 marks for assessment or exclusion from the unit.

**Project demonstrations (15 minutes) will be held during normal lab times in Week 8 (April 14 to 17)** as published on the cits1001 schedule web page. During the demonstration you will be asked to run your MyDemo, to answer questions about your code, and make modifications to your code.

The tutors will have sign up sheets to book a demonstration the labs from week 6. You must sign up for a session and attend your demonstration at the correct time to qualify for the 15 marks. It will not be possible to hold your demonstrations outside your booked time, except in exceptional extenuating circumstances which must be reported through the correct Faculty channels. Students who miss their demonstration will be awarded 0 marks for this part of the assessment.

## Assessment Criteria

This project will be marked out of 50 using the following criteria:

### Correctness (/20)

(Have all methods in Collage.java been implemented? Do they work as specified by the provided JUnit tests in the CollageTest class? )

### Style (/5)

(Is the code clearly laid out with correct indentation and good choice of variable names? Have good choices been made for programming constructs and patterns ? Is every class, method and field commented as it should be ? Have the author and version tags been completed in your classes ?)

### Demonstration (/15)

(Does the MyDemo class run and successfully illustrate the correctness of your code? Can you explain your Java code and make minor changes to it ? )

### Challenge Tasks (/10)

(Assess the quality and quantity of the code written for the challenge task. Does the Challenge.txt file contain a clear summary of what has been done and how?)

*Good luck!*

*Rachel Cardell-Oliver*

*CITS1001 Unit Coordinator*

*27 March 2014*