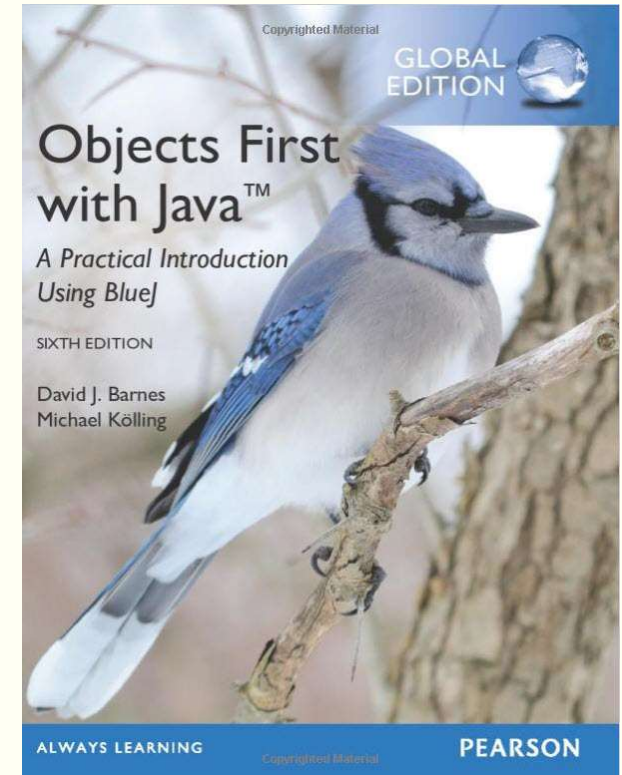


CO452 PROGRAMMING CONCEPTS

By Nicholas Day & Derek Peacock



Outline of Lesson

- Introduce the Module
- Introduce the Tutors
- Introduce Our Approach
- Assessment & Marking
- Demonstrate using Git
- Demonstrate using BlueJ
- Start the weeks Assessment
- Demonstrate Testing and Markdown

Learning Outcomes



Analyse a **simple requirement** in a structured manner in order to establish a strategy to solve the current problem



Design, document, implement and test reliable, **maintainable** programs as solutions to simple problems



Use structured techniques of design and implementation and **good documentation practice**



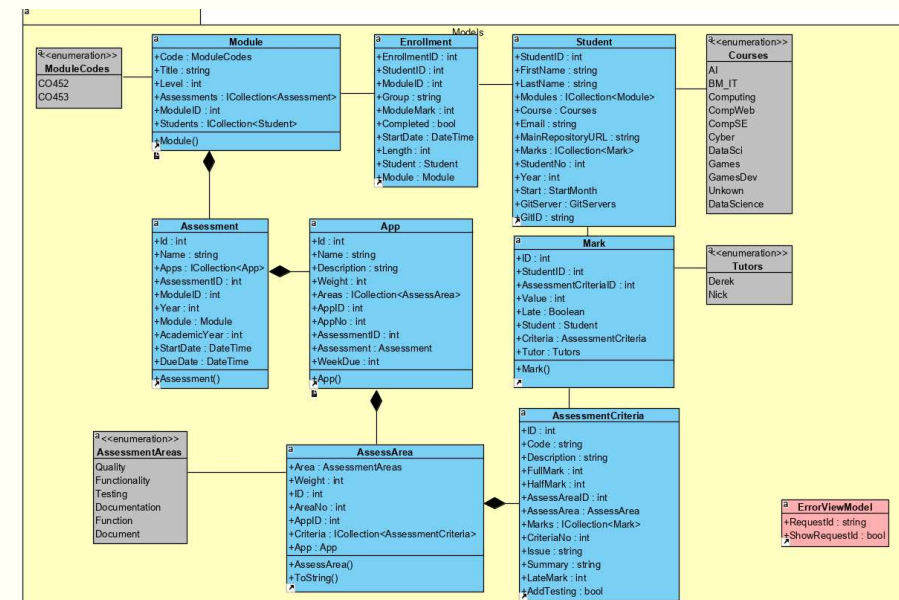
Make effective use of **software development tools** when implementing fit-for-purpose solutions

Nicholas Day – Short Bio

- Degree in Computing, PGCert in Teaching and Learning in Higher Education, PhD in Computer Science Education Research (CSEd)
- 11 years of programming experience; mainly in C++, C#, and Java
- 8 years of teaching experience; ranging from teaching introductory programming courses, to data structures and algorithms
- Post-PhD, the continuation and application of my research has led to the adoption of the flipped classroom approach – involving both the modification of modules and the production of video resources.

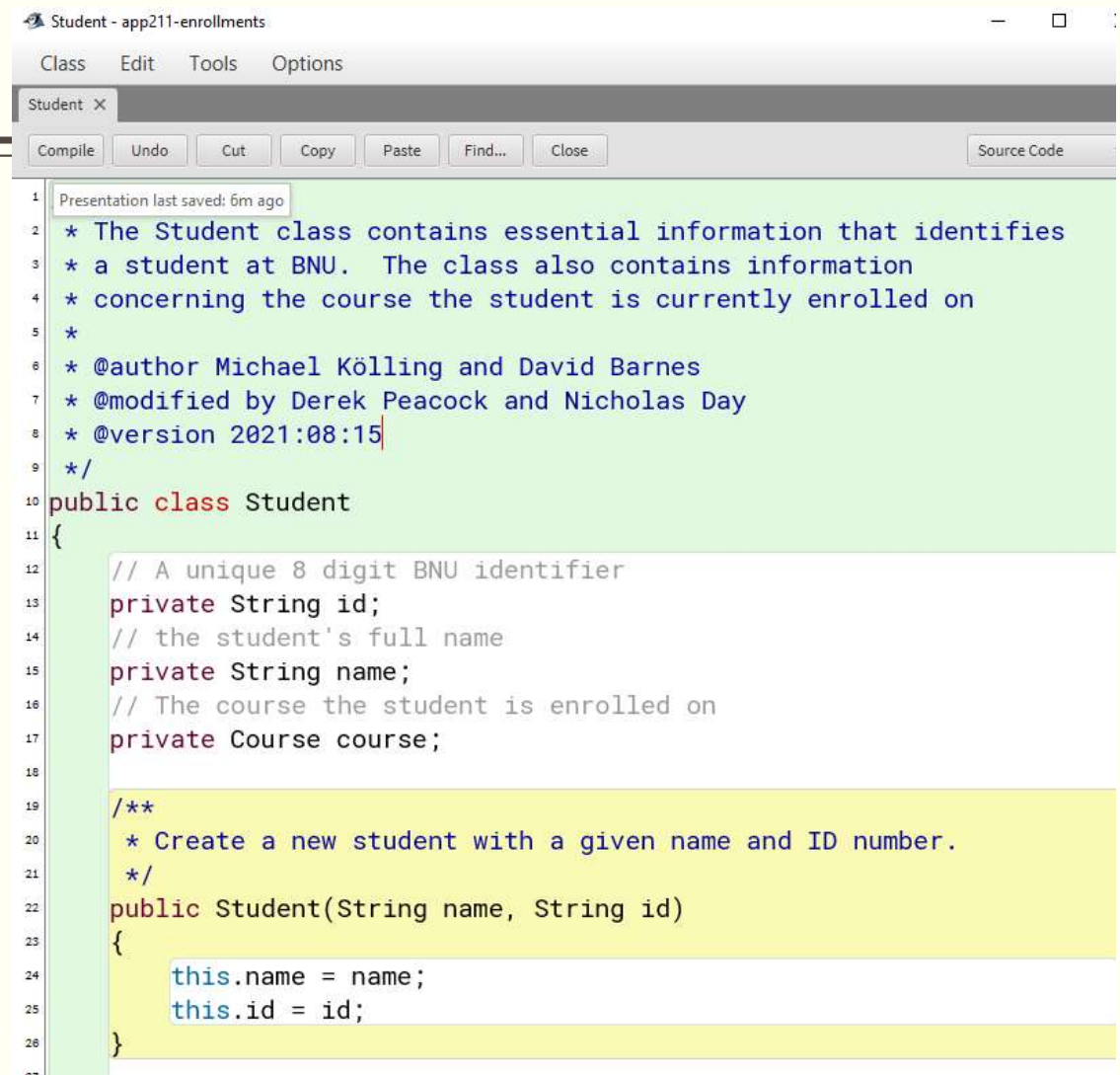
Derek Peacock – Short Bio

- Degree in Botany, PhD Plant Biochemistry
- 8 Years: Lecturer in Biochemistry,
 - Universities Durham, Sussex and Leicester
- 7 Years: Lecturer in Computing, (le.ac.uk)
- 28 Years: Lecturer in Computing (5 FE colleges)
- 50 Years Programming in 23 languages
- 38 Years Database Design (5 DBMS)
- 22 Years using CASE tools (UML)



Our Approach

- Professional Best Practice (see Code Complete & Clean Coder)
- Current In-Demand Job Skills
- Readable, simple and maintainable code
 - Good Names
 - Simple Methods
 - DRY (Do Not Repeat Yourself)
- Sound architecture
- Merciless Refactoring and Review



The screenshot shows a Java IDE window titled "Student - app211-enrollments". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar contains buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". A "Source Code" button is also visible. The code editor displays the following Java code:

```
1 Presentation last saved: 6m ago
2 * The Student class contains essential information that identifies
3 * a student at BNU. The class also contains information
4 * concerning the course the student is currently enrolled on
5 *
6 * @author Michael Kölling and David Barnes
7 * @modified by Derek Peacock and Nicholas Day
8 * @version 2021:08:15
9 */
10 public class Student
11 {
12     // A unique 8 digit BNU identifier
13     private String id;
14     // the student's full name
15     private String name;
16     // The course the student is enrolled on
17     private Course course;
18
19     /**
20      * Create a new student with a given name and ID number.
21      */
22     public Student(String name, String id)
23     {
24         this.name = name;
25         this.id = id;
26     }
27 }
```

CO452 2021 – Content

- Java Programming Concepts
- Object-Oriented Focus – BlueJ Book
- Quality rather than Quantity
- Professional Tools -> IntelliJ, Java, Git, GitHub, Wiki pages & Visual Paradigm
- Assessment which simulates **agile** professional work practices.

CO452 Blended Face-to-Face & Online Lessons

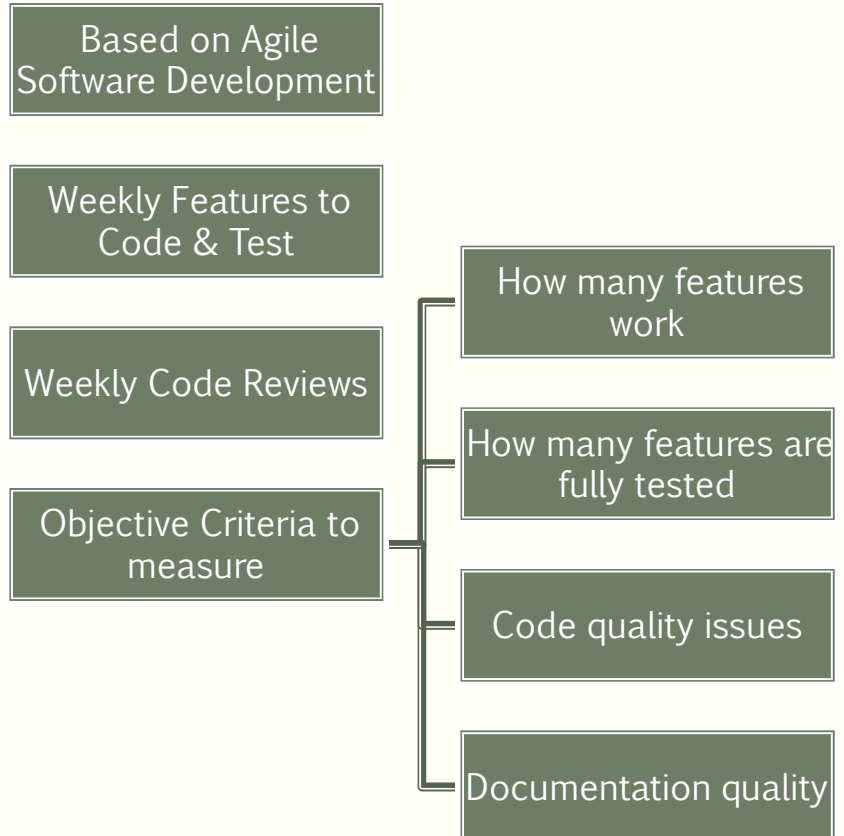
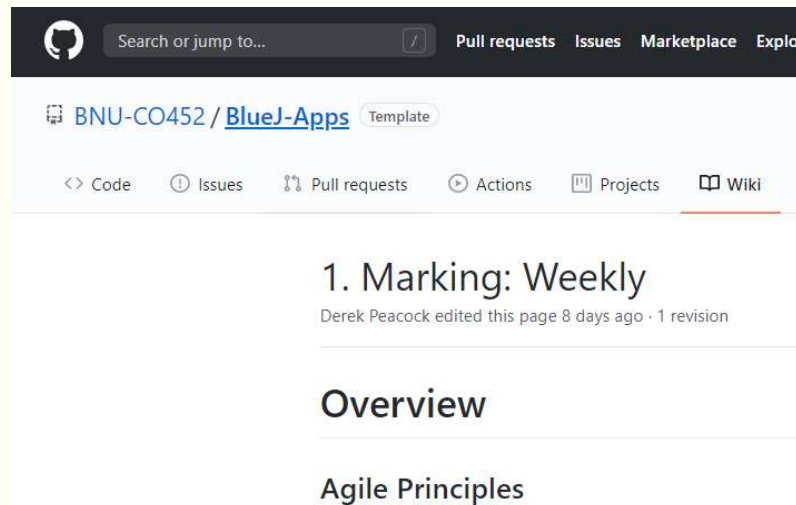
- Each week the pre-lesson Video & the BlueJ Book provide input
- The scheduled lessons are practical
 - Tutor demonstrations
 - Working through examples in one chapter
 - Getting started on the assessment
 - Students can/should help each other
 - Call on tutor when you get stuck
 - Tutor reviewing your weekly code
- Online 1:1 Help (Derek)
 - Can request 10 minute online meeting
 - Help available in set periods (times tba)
- Face-to-Face 1:1 Help (Nick)
 - Can request 10 minutes (times tba)

New Objective Assessment

This is explained more fully in the key GitHub template wiki you all must use

<https://github.com/BNU-CO452/BlueJ-Apps/wiki/Agile-Marking>

[And YouTube video](#)



Objective Marking System

- 6 Apps (Applications)
- 6 Deadlines
- 50% - Testing
- 30% - Quality (Clean Coding)
- 20% - Documentation

BNU Code Marking System

This system defines assessment marking criteria for modules that involve coding applications (Apps) using C# or Java.

Currently CO452 App02 Tickets has been filtered out by setting the due week to zero.

Learn about [building Web apps with ASP.NET Core 5.0 MVC](#).

Listing of CO452 Apps

PK	Module	App	Description	Weight
2	CO452	App21-01 Enrolments	Students enrolling on a course	10%
3	CO452	App21-02 Grades	Converting module marks into grades	15%
4	CO452	App21-03 Products	Maintaining a list of products for sale	15%
5	CO452	App21-04 Stock	Adding a user interface to the products app	20%
15	CO452	App21-05 Songs	A time limited application	20%
6	CO452	App21-06 Zuul	Colossal Adventure type of console game	20%
TOTAL WEIGHT =				100%

Simulation of Agile Practice

- Each App is like a “sprint” of 2-3 weeks.
- Working code produced by the deadline is the top priority.
- Simple quality code is better than “clever” complex code.
- If your App is late your marks for the app could be limited (40%) .
- If you are late you will not be ready for App05
- App05 must be completed in 3 hours in week 10

CO452 Module Scheme 2021

The core text book for module CO452 is - Barnes, D. and Kolling, M. (2017),
Objects-first in Blue!: A Practical Introduction to Java, 6th ed. Pearson

Week	Begin	Subject	App Start	Deadline
01	27 Sep	Classes and Objects part 1: Variables, Constants, Methods, Parameters, Output	app21-01-enrollments	
02	04 Oct	Classes and Objects part 2: Extending last week with more methods?		
03	11 Oct	Selection: if and switch, Enumerations	app21-02-grades	app21-01-enrollments
04	18 Oct	Iteration: for, for-each, while + ArrayLists		
05	25 Oct	app21-02-grades Workshop		
06	01 Nov	Collections: Another list?	app21-03-products	app21-02-grades
07	08 Nov	String lecture + app21-03-grades		
08	15 Nov	Main, User Interfaces and Input	app21-04-stock	app21-03-products
09	22 Nov	Arrays? Helpers? + app21-04 Workshop		
10	29 Nov	app21-05-songs		app21-04-stock

Testing – Limits your mark!

- If only 50% of the apps features work then the maximum mark = 50%

- In week 1 there are 14 tests to complete most on code you were given.
- Each test might only take 1 minute but must be recorded and put in your wiki.
- In the real world late delivery has serious penalties.
- Apps delivered late may only get 40% maximum.
- You can have a week to make improvements if on time.

Listing of Assessment Criteria

CO452 App211 Enrolments

Testing

Change Selection

Add Assessment Criteria


App211 Enrolments: Testing


PK	App	Area	Criteria No	Category	Summary	Description	Mark	Actions
348	App211	Testing	1	Object-Bench	Create a student object	with your name and id	4	Edit Delete
352	App211	Testing	2	Object Bench	Print student object		2	Edit Delete
350	App211	Testing	3	Object-Bench	Create a course object	with your course title and code	4	Edit Delete
351	App211	Testing	4	Object Bench	Print course object		2	Edit Delete
349	App211	Testing	5	Object-Bench	Enrol student on course		2	Edit Delete
353	App211	Testing	6	Object Bench	Print student's course	use student object	4	Edit Delete

Today's Practical - GitHub

1. Create a GitHub account (personal email)
2. Goto <https://github.com/BNU-CO452/BlueJ-Apps/wiki>
3. Create a shortcut to this wiki
4. Go to <Code> in the menu
5. Create your public **BlueJ-Apps** repository by clicking on “Use this template”
6. Create your wiki home page by copying the Example Home Page (slides)
7. Give your tutor (bnu-cms)
 1. Your full name
 2. Your GitHub id

GitHub Profiles

 Search or jump to... Pull requests Issues Marketplace Explore



Antony Peacock
Twon

Member of the ISO C++ Standardisation Committee and the BSI C++ Panel. High performance computing specialist focusing on mathematical software.

Unfollow ...

8 followers · 66 following · 70

London

Overview Repositories 34 Projects Packages

Twon / README.md

Hi there, I'm Antony 🐼

I'm a software engineer who's passionate about contributing to and improving the open-source C++ ecosystem. I've worked in video games, high performance computing, quantitative analytics, electronic trading and low-latency trading system.

Antony Peacock's GitHub Stats

☆ Total Stars:	8
🕒 Total Commits (2020):	308
🔗 Total PRs:	120
🔔 Total Issues:	77
💻 Contributed to:	9


Visitor count

0 0 0 0 0 4 0


Antony Peacock
Twon

Member of the ISO C++ Standardisation Committee and the BSI C++ Panel. High performance computing specialist focusing on mathematical software.

Pinned

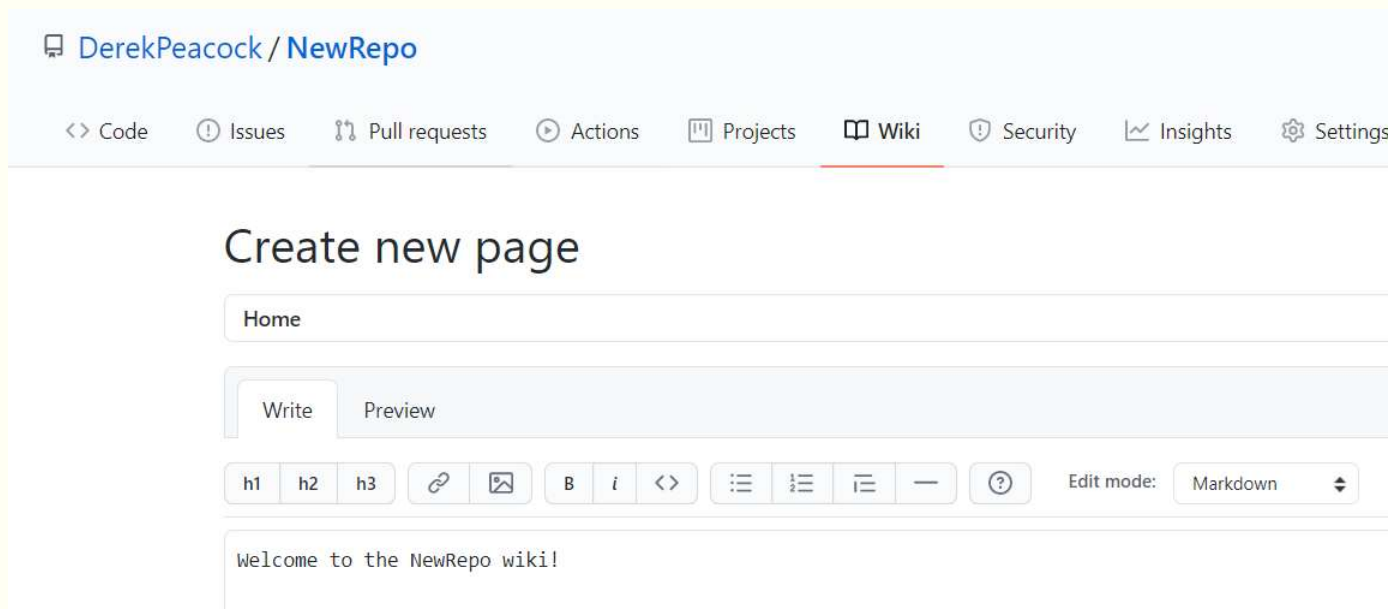
 **Morpheus**

Work in progress: Cross Platform Game Engine

 **polymorphic_value**

Forked from jbcoc/polymorphic_value

How to Update your Wiki (using Markdown)



- Create a new wiki page called **Home**
- Copy the example student home wiki page
- See markdownguide.org
- See wiki video

<https://github.com/BNU-CO452/BlueJ-Apps/wiki/Student-Home-Page.md>

Student Name: Home Page

Student ID: {21900000}

[Watch this Video on How to Create Your Wiki Pages](#)

Course: {Student Course}

A paragraph of your interests in computing and why you enrolled in your course

CO452 Lesson : {Fri 9:00 Nick}

1. [App21-01-Enrollments](#)
2. [App21-02-Grades](#)
3. [App21-03-Products](#)
4. [App21-04-Stock](#)
5. [App21-05-Songs](#)
6. [App21-06-Zuul](#)

Student Home Page

Write

Preview

h1

h2

h3



B

i

<>



Edit mode:

M

```
# Student Name: Home Page
## Student ID: {21900000}
[Watch this Video on How to Create Your Wiki Pages](https://youtu.be/pGLc8avz4e8)
## Course: {Student Course}
A paragraph of your interests in computing and why you enrolled in your course
## CO452 Lesson : {Fri 9:00 Nick}
1. [App211-Enrollments]()
2. [App212-Grades]()
3. [App213-Products]()
4. [App214-Stock]()
5. [App-215-Zuul]()
```


Application Documentation – GitHub Wiki Pages

- This will include:-
 - Brief description of the app
 - Required features
 - Design
 - Testing
 - Evaluation
- Design is created by BlueJ & UML
- Description and Features can be copied from the wiki template
- Testing & evaluation needs adding to your repository & wiki

App-211 Student Enrolments

Description

This app is modelled on the way BNU (Bucks New University) works, and this is similar to most u
See [BSc Awards](#)

A Student can enrol on a Course. The Course class stores the course code and the title of the co
stores the students ID and name. Java code already exists for most of the features. but this code

Codes or IDs are commonly used whenever you have many items (students, courses, modules) v
identical names. They are an essential part of the way relational database work. They have to be
should be easy to recognise.

Testing Requirements Stage 1

Document your testing by selecting **View -> Show Terminal** in BlueJ and then Select **Options ->**
the testing is complete save to a **txt** file as "**Tests01.txt**" in your App-211 folder. Ensure that Clea

1. Create a new Course object called "**computing**" for example
2. Set the BNU Course **code** (see below) and the course **title** to your own course title for exam
3. Print the **course** details.
4. Create a new student object with your **name** and your student **id**.
5. Print out the details the student object.
6. Enrol the student on the Course.
7. Print out the details of the student's course
8. Screen shot your object bench at the end of testing and save the file in your app211-enroln
9. Save your terminal window output to a **txt** file as "**Tests01.txt**" in your App-211 folder.

Course Code	Course Title
-------------	--------------

CO452 Module Index

[Module Scheme](#)

2021 Apps Requirements

[App21-01-Enrolments](#)
[App21-02-Grades](#)
[App21-03-Products](#)
[App21-04-Stock](#)
[App21-05-Songs](#)
[App21-06-Zuul](#)

Videos

[Pre-lesson Videos](#)
[Recorded Lessons](#)

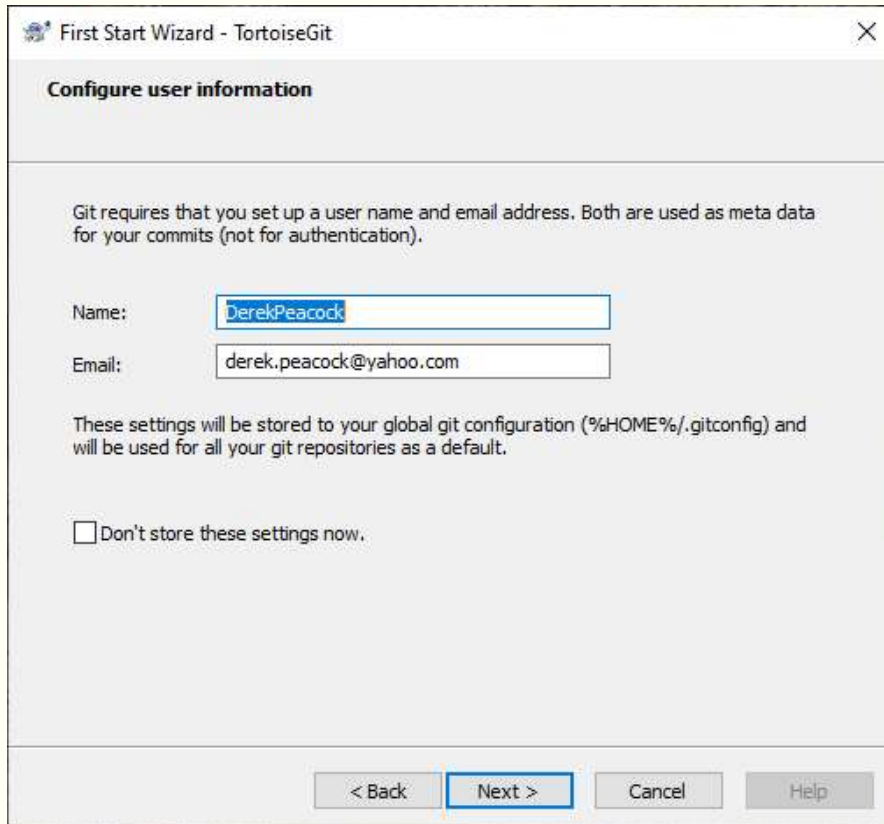
Resources

[Git and GitHub](#)
[Example Home Page](#)
[Wiki Documentation](#)
[BSc Awards](#)
[BlueJ Extensions](#)
[BlueJ Issues](#)
[Java SDK](#)
[Java IDEs](#)
[Coding Principals](#)
[MTA Exams](#)

Module Marking

[Agile Marking](#)
[Testing Marking](#)
[Quality Marking](#)

Demonstration of Tortoise Git -> Cloning



The screenshot shows the 'First Start Wizard - TortoiseGit' window. The title bar includes a close button (X). The main heading is 'Configure user information'. Below this, a message states: 'Git requires that you set up a user name and email address. Both are used as meta data for your commits (not for authentication)'. There are two input fields: 'Name:' with the text 'DerekPeacock' and 'Email:' with the text 'derek.peacock@yahoo.com'. Below the fields, a message says: 'These settings will be stored to your global git configuration (%HOME%/.gitconfig) and will be used for all your git repositories as a default.' There is a checkbox labeled 'Don't store these settings now.' which is currently unchecked. At the bottom, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'. The 'Next >' button is highlighted with a blue border.

- Run Tortoise First Start, enter your name and email address
- Or right click TortoiseGit -> Settings
- Create an empty folder called repos
- Goto GitHub and copy the remote repository HTTPS URL
- In the repos folder right click and select Tortoise Clone, paste in the URL

Download & Install BlueJ (at home)

Documentation

BlueJ

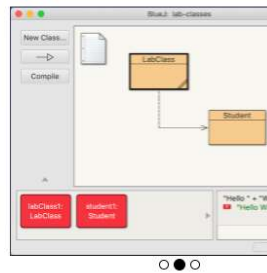
A free Java Development Environment designed for beginners, used by millions worldwide. [Find out more...](#)

"One of my favourite IDEs out there is BlueJ"
— James Gosling, creator of Java.

Created by



Supported by **ORACLE**



Download and Install

Version 5.0.2, released 6 August 2021 (fixes crash on launch and minor bugs, [and more](#))

Windows



Requires 64-bit Windows, Windows 7 or later. Also available: Standalone zip suitable for USB drives.

Mac OS X



Requires OS X 10.11 or later.

Ubuntu/Debian



Requires 64-bit, Debian buster or Ubuntu 18.10 or later.

Other



Please read the [Installation instructions](#). (Works on most platforms with Java/JavaFX 11 support).



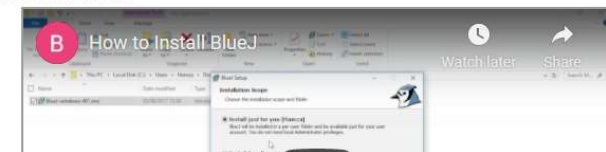
BlueJ: Short Video Tutorials

We have provided short video tutorials to support beginner in some common topics on how to use BlueJ.

Contents

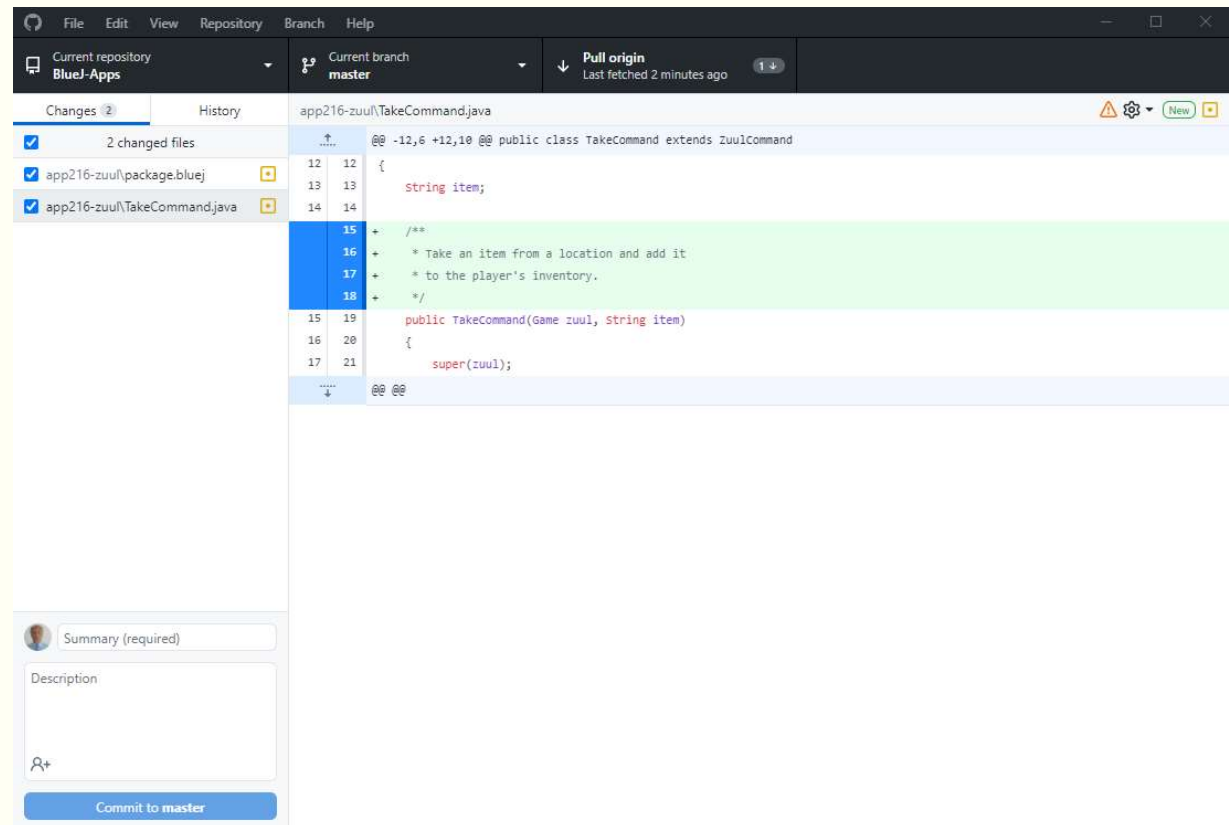
1. [How to Install BlueJ](#)
2. [How to Change the Interface Language](#)
3. [How to use the Debugger](#)
4. [How to Reset the JVM](#)
5. [How to Open and Save Projects](#)

1. How to Install BlueJ



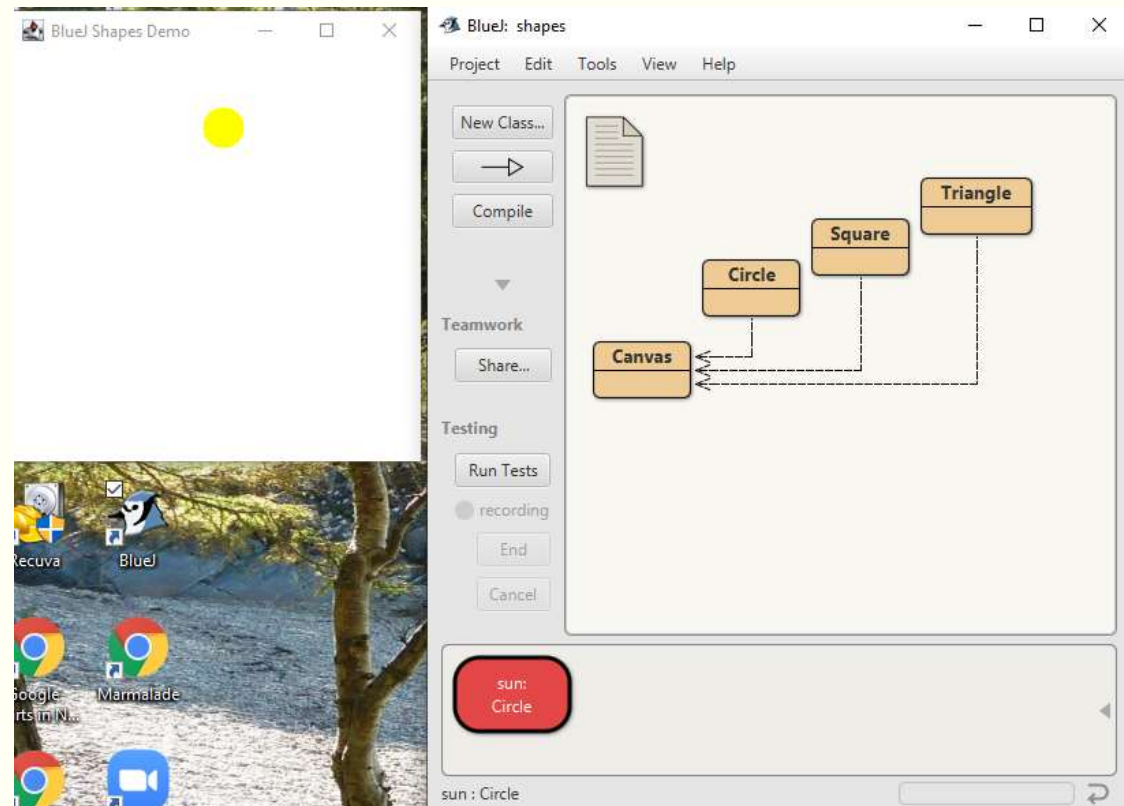
Download and Install GitHub Desktop (at home)

- See CO452-BNU\BlueJ-Apps readme
- Easy to use git gui
- Integrates with GitHub
- Also see BlueJ-Apps wiki



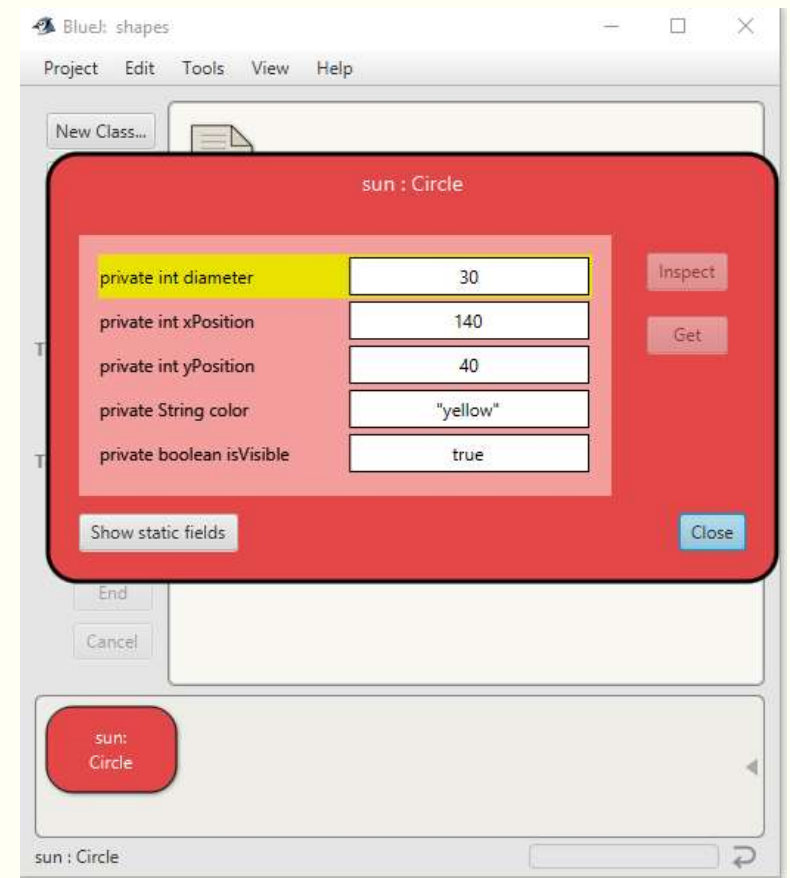
Practice – BlueJ Objects Part 1

- Open the shapes project.
- Create a new circle **object** called sun.
- Right click on the sun and call the `makeVisible()` **method**.
- Call the `moveRight()` method.
- Call the `moveUp()` method
- Call the `moveHorizontal(100)`
- Call `changeColor("yellow")`
- Values in () are called **parameters**



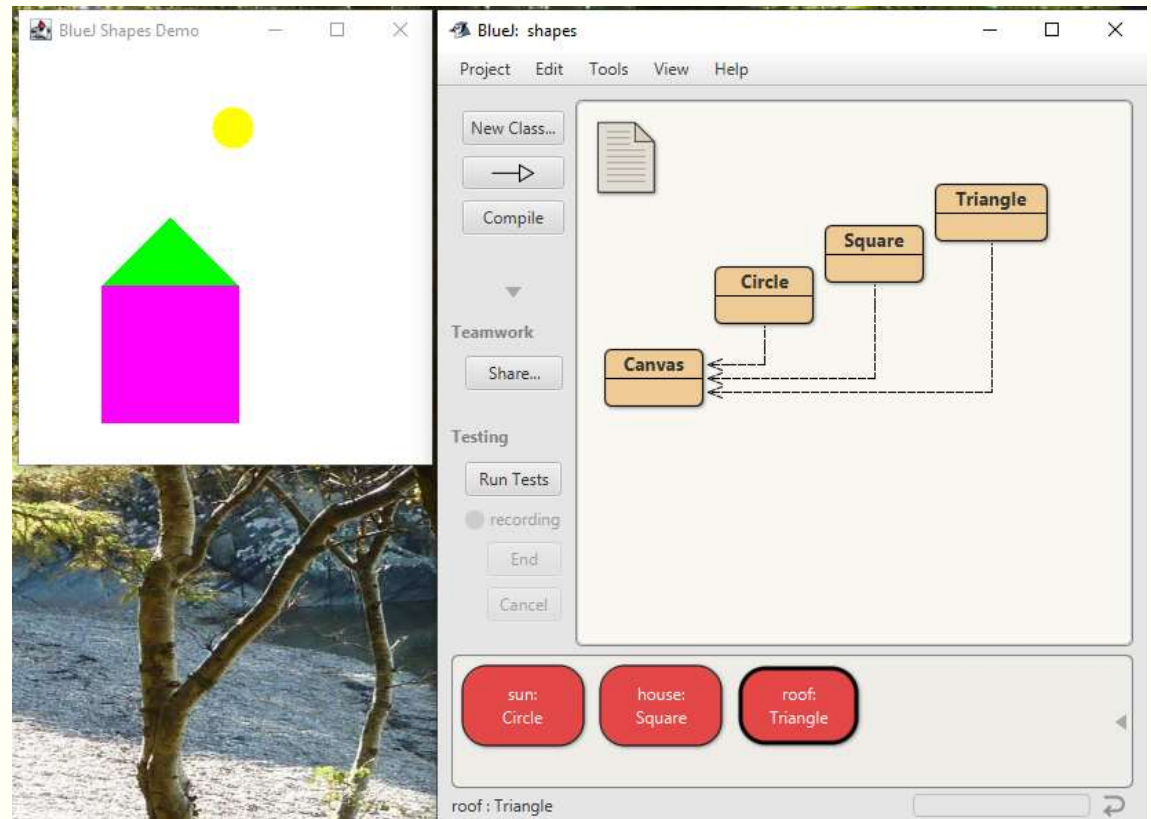
Practice – BlueJ Objects Part 2

- Inspect the sun object
- It contains **data values** of various types
- Int (whole numbers), String (text values), boolean true/false.
- diameter is a **variable/attribute**.
- Create a square house
- Create a triangle roof
- Use methods to arrange them into a picture.

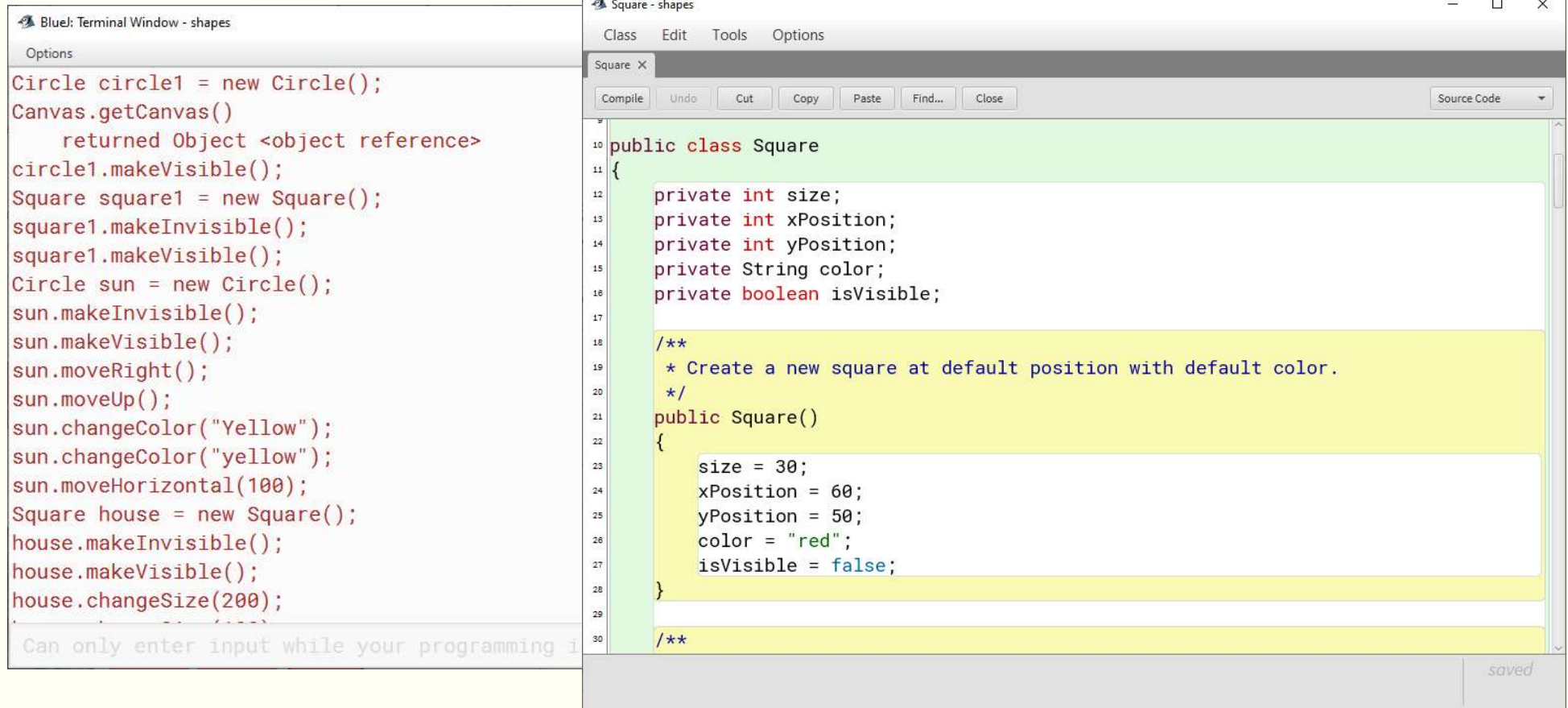


Practice – BlueJ Objects Part 3

- Only colors are yellow, green, blue, black, magenta and red.
- Select **Show Terminal** from the view menu.
- Make sure Show Method Calls option is set to on.
- What you will see is the Java code needed to call the methods.
- Double click on the Square class and look at the Java code.



Java Code



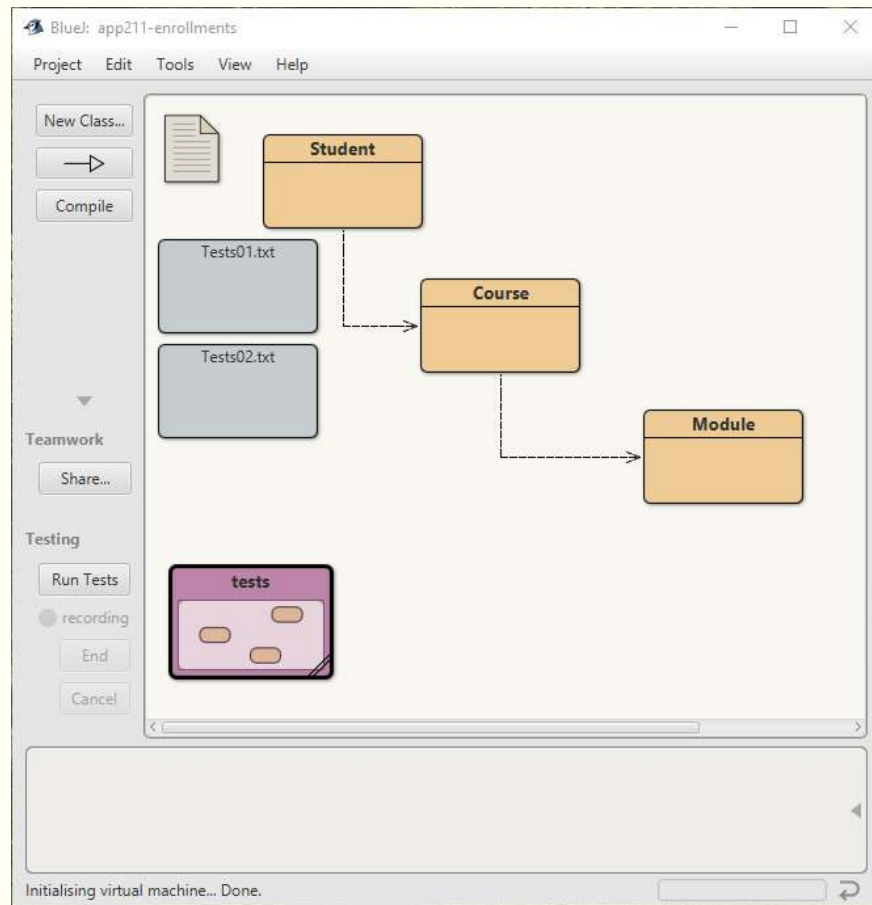
```
BlueJ: Terminal Window - shapes
Options
Circle circle1 = new Circle();
Canvas.getCanvas()
    returned Object <object reference>
circle1.makeVisible();
Square square1 = new Square();
square1.makeInvisible();
square1.makeVisible();
Circle sun = new Circle();
sun.makeInvisible();
sun.makeVisible();
sun.moveRight();
sun.moveUp();
sun.changeColor("Yellow");
sun.changeColor("yellow");
sun.moveHorizontal(100);
Square house = new Square();
house.makeInvisible();
house.makeVisible();
house.changeSize(200);

Can only enter input while your programming i
```

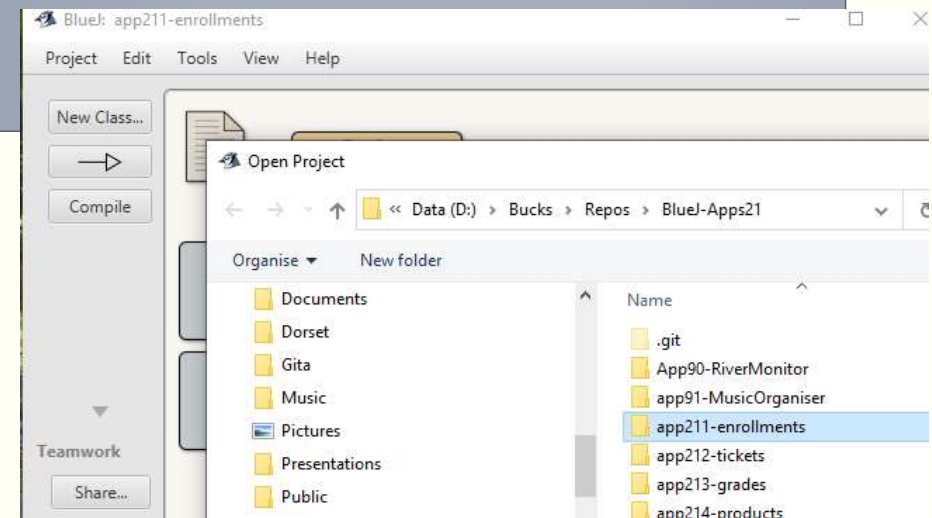
```
Square - shapes
Class Edit Tools Options
Square X
Compile Undo Cut Copy Paste Find... Close Source Code
10 public class Square
11 {
12     private int size;
13     private int xPosition;
14     private int yPosition;
15     private String color;
16     private boolean isVisible;
17
18     /**
19      * Create a new square at default position with default color.
20      */
21     public Square()
22     {
23         size = 30;
24         xPosition = 60;
25         yPosition = 50;
26         color = "red";
27         isVisible = false;
28     }
29
30     /**
31
32     */
33 }
```

saved

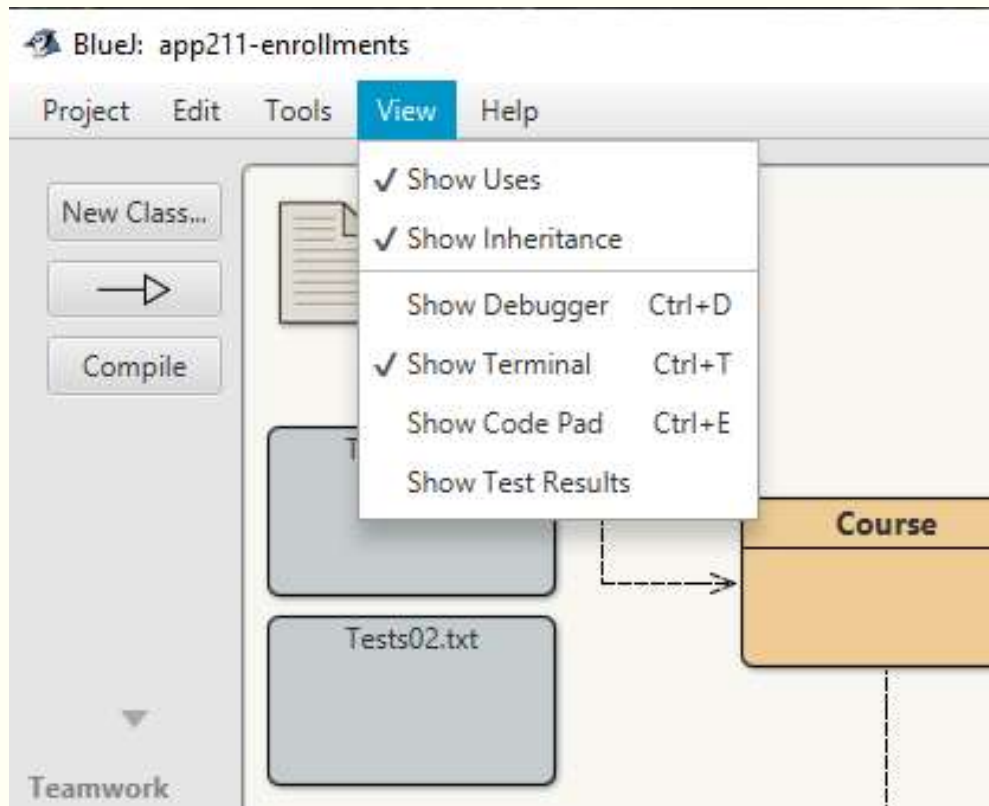
Open the **app21-01-enrollments** Project



- Open BlueJ
- Open Project and find the **app211-enrollments** project folder
- Look at assessment details
<https://github.com/BNU-CO452/BlueJ-Apps/wiki/App-211>



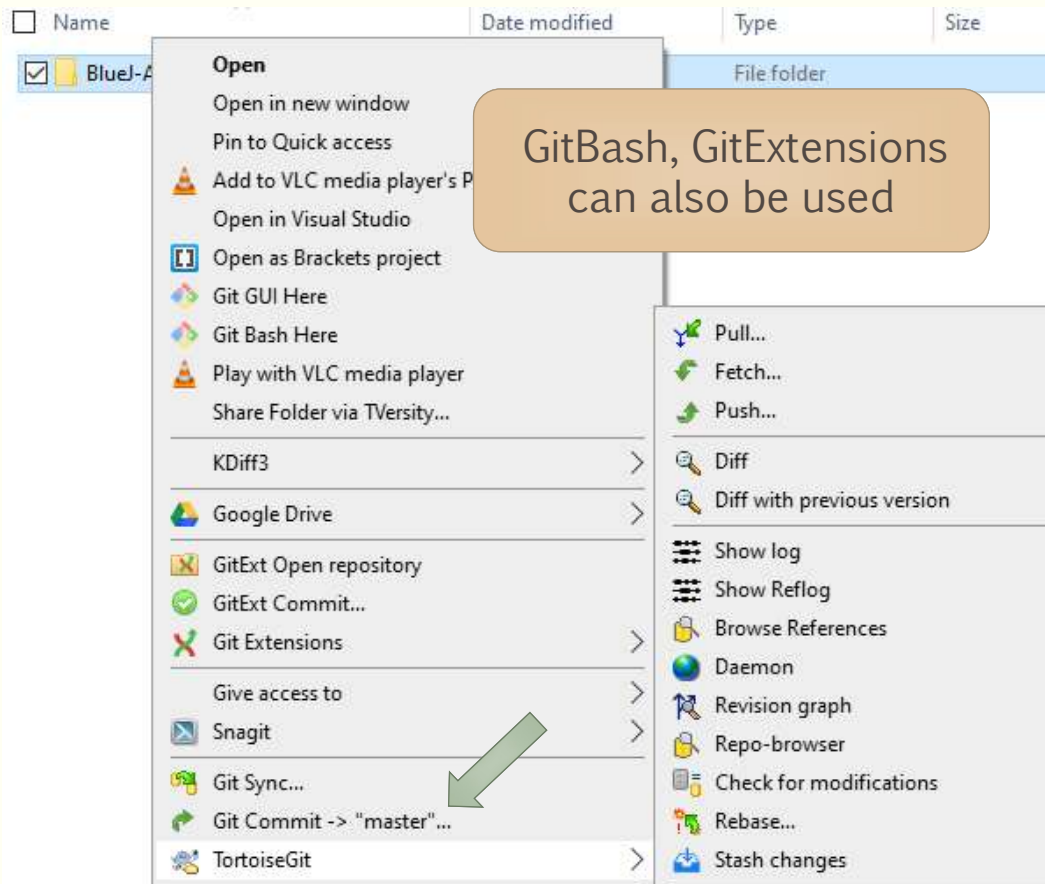
How to document Required Testing (wiki App21-01-Enrolments)



- Make sure that the terminal view is selected
- Create a student object “gita”
- Create a course object “computing”
- Enrol a student on the course
- In the terminal window save the file as Tests01.txt to the app211-enrollments folder
- Commit & Push the changes

See wiki on Git

Tortoise Git -> Making Changes



Edit the **Student** class and update the class comment and add your name as modified by

Save the changes and **compile**

Right click on the repository folder and **commit** the changes (can be done four ways)

Push up the changes up to the remote repository

At home for first time **clone** you remote repository

Next time **pull** down the changes when you start work using GitHub Desktop.