COMP 1409 - Introduction to Software Development 1

Mike Mulder, P. Eng.

Agenda

- Introductions
- Course Logistics
- Software Development Overview
- Lesson 1
  - BlueJ Development Environment
  - Definitions
  - Simple Java Class
- Lab 1

Introductions - Instructor

- Mike Mulder, P.Eng, CSD/CSM/CSPO
- My Interests:
  - Programming (Java, Python, Ruby)
  - Software Requirements and Architecture
  - Software Project Management
- Office Hours – By Appointment
  - I am available before and after class
- E-mail: mmulder10@bcit.ca

Introductions - Instructor

Introductions – Each Other

Turn to the person next to you and interview them about:

- Why they are taking this course?
- Do they have any previous programming experience? If so, in what language(s)?
- What is their work or educational background?

Then please introduce each other to the class.

Before We Begin…

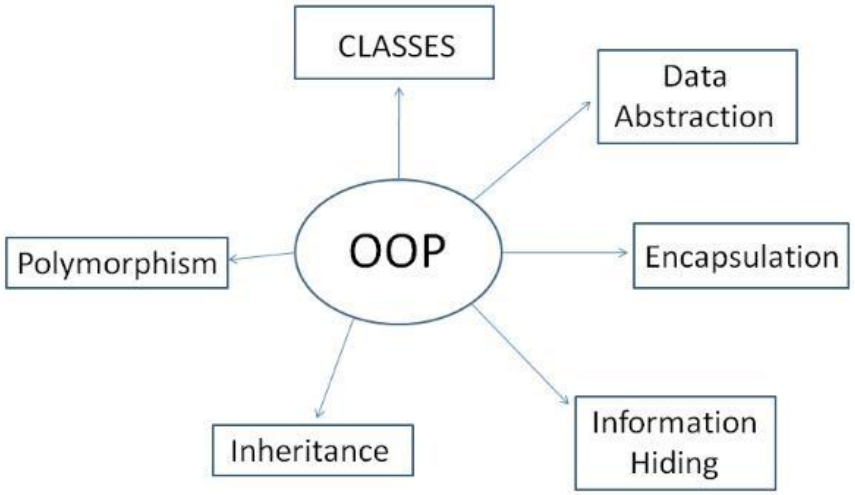
- Please make sure you can login to the lab machines using your BCIT student number (A0 number) and password
  - Let me know if you cannot login or do not see a BlueJ icon in the top left corner of the desktop.

You are welcome to use your own laptop with BlueJ installed for the class.
You will need BlueJ installed on your personal computer for some labs and all assignments.

https://www.bluej.org/

COMP 1409

- Introduction to Software Development, focusing on:
  - Object Oriented Programming (OOP)
  - The Java Programming Language



Goal and Outcomes

Goals:

- To provide an understanding of object-oriented programming concepts.
- To prepare students for follow-on courses in software development.
- To use OOP concepts and code to solve problems.

Learning Outcomes:

- Explain object-oriented programming concepts including: object, class, method, and encapsulation.
- Explain abstraction and modularization in object-oriented programming.
- Describe data types used in programming.
- Explain the use of variables and constants.
- Identify and use control structures.
- Use correct syntax and documentation standards.
- Read class library interfaces.
- Write simple programs in Java.
- Test and debug simple programs in Java.

Course Evaluation

Criteria	%	Comments
Labs	20	Some labs are in class; others are done at home.
Quizzes	30	There is a quiz at the beginning of every session except the first and the last.
Assignments	20	There are three take-home assignments which must be completed in order to pass this course.
Final Exam	30	You must score at least 50% on the final exam to pass this course; half of the exam is theoretical (paper); half is practical (computer).

Passing Grade: 60%
Late assignments will not be accepted for marking unless the student makes arrangements with the instructor before the assignment is due. Assignments must be done on an individual basis unless otherwise specified by the instructor.

There are three take-home assignments worth a total of 20%. The take-home assignments will be downloaded and submitted through the course Learning Management System.

Students are encouraged to work in groups to develop peer to peer communication and support. However, **each student must hand in their own individual work** (not copies of the same assignment). Plagiarism and other forms of cheating will not be tolerated. Assignments are to be completed by each student on an individual basis unless stated otherwise. Any form of **plagiarism will result in a grade of ZERO** for the first instance. Any subsequent instances of Academic Misconduct will meet with harsher penalties. These penalties may include failing the course and/or removal from the program.

Course Topics

Week	Date	Topics	Comments
1	Sept. 8	Bluej Definitions: Classes, objects, data types, state, methods Naming conventions A simple java class	
2	Sept. 15	Comments Visibility modifiers Methods: return types and parameters if statements null Constructors Parameters Assignment statement Logical operators Relational operators	Quiz 1
3	Sept. 22	Accessors Mutators Console output String concatenation public static void main()	Quiz 2
4	Sept. 29	Logical operators Local variables Static variables and methods Constants	Quiz 3

Course Topics

Week	Date	Topics	Comments
5	Oct. 6	Thanksgiving – No Class	
6	Oct. 13	Arithmetic operators Overloading Switch/case	Quiz 4
7	Oct. 20	Composition (object interaction and external method calls)	Quiz 5, Assignment 1 Due
8	Oct. 27	References; identity versus equality null (again) this Debugging techniques	Quiz 6
NOTE: Course Withdrawal Deadline Please inform your instructor that you are dropping this course. You must also fill out and submit the **'REQUEST TO WITHDRAW FROM A PART-TIME STUDIES COURSE'** before session 8 or else you will receive a failing grade on your academic record**.**			
9	Nov. 3	Arrays while loops	Quiz 7
10	Nov. 10	Remembrance Day – No Class	
11	Nov. 17	More arrays for loops	Quiz 8, Assignment 2 Due
12	Nov. 24	ArrayList class Enhanced for (foreach) loop	Quiz 9

Course Topics

Week	Date	Topics	Comments
13	Dec. 1	Iterators	Quiz 10
Course Evaluation: To be conducted online during Session 11 prior to the class break.			
14	Dec. 8	Final Exam	Assignment 3 Due

Typical Class Structure

- Quiz
  - Based on the previous week's lecture
- Mini Lecture
  - On the topics for the week
- Lab
  - Hands on practice on the week's topics
  - Both in-class and take-home components

Course Resources

This course uses on-line resources to be provided by your instructor, no text book is required.

Textbook (Optional):

- Java Early Objects
  Adrian Lizarraga
  zyBooks
  http://www.zybooks.com/catalog/java-early-objects/

Software:

- Java Platform (JDK) from Sun Microsystems, downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html
- BlueJ interactive development environment, downloaded from www.bluej.org

Learning Hub (also known as D2L)

https://learn.bcit.ca

Includes:

- Course announcements
- Lecture slides
- Dropbox for assignments

Please make sure you can login and access the COMP 1409 page.

Attendance

PTS Attendance Policy

- Attendance in lectures and labs is mandatory and recorded for all lessons in this course.
- In case of illness or other unavoidable cause of absence, the PTS student must communicate as soon as possible with his/her Instructor indicating the reason for the absence.
- Prolonged illness which causes the PTS student to miss 20% or more of the lessons will require a BCIT -approved medical certificate submitted to the department, substantiating the reason for the absence.
- Excessive absence of 20% or more may result in failure or forced withdrawal from this course.

Student Resources

Please review these resources before the second lesson.

- **Welcome to Computing PTS**
  - https://youtu.be/C0gtCxVO6f0
- **Computing Part-time Studies Student Guide**
  - http://www.bcit.ca/files/cas/computingparttime/pdf/computing_pts_student_guide.pdf

Please Note: By attending this course you agree to read this Student Guide.

Software Development Overview

What is Software Development?

Also known as software programming or coding

Software Development:

Building of software applications, frameworks or components.

Activities include specifications, design, programming, documenting, testing, bug fixing, maintenance.

Types of software include web applications, mobile applications, desktop applications, operating systems, embedded software, etc.

Programming Languages

Low Level Languages

- Assembly Language – programming in a language native to the machine you are programming for.
- No longer used except in specialized cases.

High Level Languages

- Use abstractions to make programming more accessible and intuitive.
- Abstractions include variables, arrays, objects, complex arithmetic, Boolean expressions, functions, subroutes, loops, etc.

Programming Languages - TIOBE

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	**2**		**C**	**15.447%**	**+8.06%**
3	5	↑	Python	7.653%	+4.67%
4	3	↓	C++	7.394%	+1.83%
5	8	↑	Visual Basic .NET	5.308%	+3.33%
6	4	↓	C#	3.295%	-1.48%
7	6	↓	PHP	2.775%	+0.57%
8	7	↓	JavaScript	2.131%	+0.11%
9	-	⇈	SQL	2.062%	+2.06%
10	18	⇈	Objective-C	1.509%	+0.00%
11	12	↑	Delphi/Object Pascal	1.292%	-0.49%
12	10	↓	Ruby	1.291%	-0.64%
13	16	↑	MATLAB	1.276%	-0.35%
14	15	↑	Assembly language	1.232%	-0.41%
15	13	↓	Swift	1.223%	-0.54%
16	17	↑	Go	1.081%	-0.49%
17	9	⇊	Perl	1.073%	-0.88%

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. **The ratings are based on the number of skilled engineers world-wide, courses and third party vendors.** Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

Source: https://www.tiobe.com/tiobe-index/

Programming Languages - Java

- First released in 1995
- Simple – removes some of the complexities of older languages like C/C++
- Object Oriented
- Compiled
- Portable – compiled code works on any machine that supports a Java Runtime Environment
- Still heavily used in the software industry, especially for enterprise applications and larger teams

Programming Paradigms

- Paradigm – Describes methods of programming, typically to reduce or control complexity
- Examples:
  - Procedural Programming – Software is broken down into procedures that are executed when the program is run.
  - Functional Programming – Similar to procedural programming, where the software is broken down into functions. Functions are stateless, similar to mathematical functions.
  - Logical Programming – Program is expressed as relations, facts, rules.
  - Object Oriented Programming – Organize code to represent objects with behaviors.

Programming Paradigms – OOP

- OOP = Object Oriented Programming
- Objects are the building blocks of a program, where a program is a collection of interacting objects.
- Objects have state (i.e., data) and behavior
- Four Pillars of OOP:
  - Abstraction
  - Encapsulation
  - **Inheritance***
  - **Polymorphism***

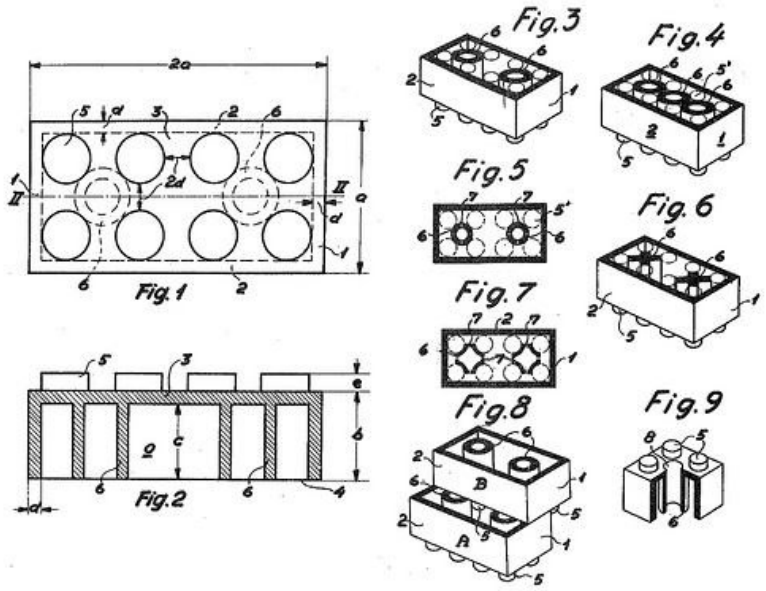
*** See COMP 1451**

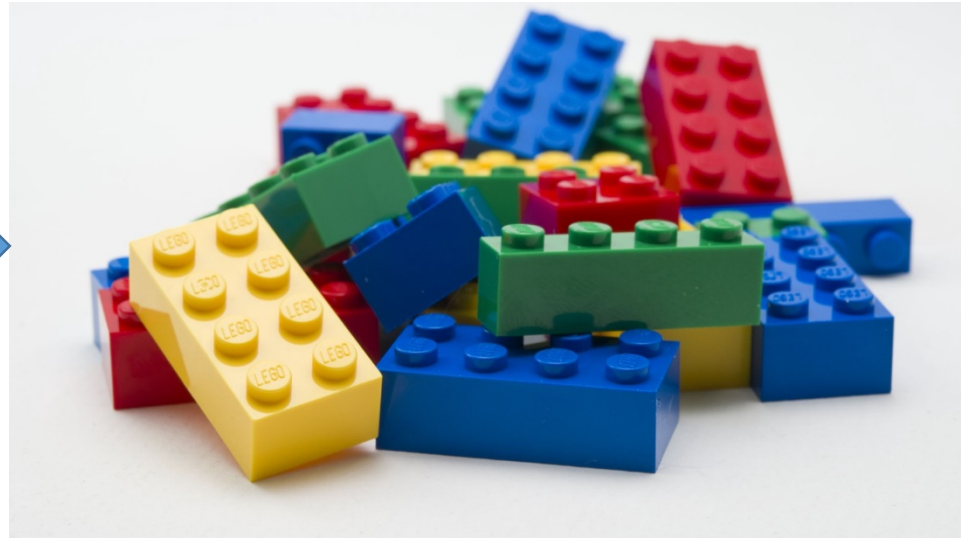
For this course, we will focus on the fundamentals of designing and building objects.

Programming Paradigms – OOP

Benefits:

- Reuse of Code – Either in the same or different programs
- Testability – Objects can be tested independently
- Extensibility – Easier to add new data or behavior to objects





COMP1409:
Lesson 1

Learning Outcomes: Lesson 1

- BlueJ
- Definitions:
  - classes
  - objects
  - data types
  - object state
  - methods
- Naming conventions
- A simple Java class

BlueJ

- BlueJ is the editor we will use to write Java
- It is a tool intended for new students
- It is not a tool for professionals
- But the compiler is the actual (professional) Java compiler
- We will start using it today.

Definitions: class

- In this course, we write classes in Java
- A Java class is a file
- The file describes a general category in terms of:
  (a) data and
  (b) behaviors
- For example: we can write a Book class, which describes all books.
- All books have a title, an author, a date written; can be opened, closed, etc….

- Another example: we can write a BankAccount class, which describes all bank accounts.
- All bank accounts have an account number, a date opened, and a balance (say, in US dollars); can transfer, withdraw, deposit, etc….

Definitions: object

- Whereas a class describes a general category, an object is one specific instance of a class.

- For example, once you have described what a Book is, and that it has a title, an author, and a date written, you can then construct Book objects. Every Book would therefore have a title, an author and a date written.

- Harry Potter is the title of a Book written in 1995 by J.K. Rowling
- Lord of the Rings is the title of a Book written in 1954 by J.R.R. Tolkien
- etc

Style

- https://google.github.io/styleguide/javaguide.html
- The goal is maximum readability and clarity.

Go to the Java style guide above, and find the following best practices:

- **Naming of classes**
- **Naming of methods**
- **Naming of local variables**
- **What is Javadoc and when is it used?**

Why is a style guide such as this important?

Naming Conventions

CamelCase – A capital is used for the first letter of each word in a compound word or phrase. The remaining letters are lower case.

lowerCamelCase – The first letter of the first word is lower case.

UpperCamelCase – The first letter of the first word is upper case.

Class Name	UpperCamelCase	Book, BankAccount
Variable Name (except constants)	LowerCamelCase	yearPublished, balance, isOverdrawn
Method Name	LowerCamelCase	getPrice, getBalance, withdraw

Definitions: data type

- Each piece of data in a class is known as an instance variable
- It may be also called a field, a data member, a property, or other names
- Each instance variable has a data type
- Instance variables should be named as nouns or adjectives; not as verbs
- For example, the title of a Book is a String
- The year in which a Book is written is an int
- The balance in U.S. dollars of a bank account is a double.

Definitions: data type

- Java has two categories of data types.
- 1. Primitive data types
  - int
  - double
  - boolean
  - char
  - (there are others too, which we do not need in this course)
- 2. Reference (or Object) types
  - String
  - Book and BankAccount are also reference types. Classes are reference types.
  - Note the Capital first letters of these data types

Example class

```
class Book{

   // instance variables:
   String          authorFirstName;
   String          title;
   int             yearPublished;
   int             numberOfPages;
   double          priceUSD;
   double          weightKg;
   boolean         originalLanguageEnglish;
   boolean         fiction;
   char            edition;
}
```

Definitions: data type

Data type	Description	Default instance variable value set by the compiler on the heap	Examples
int **(primitive)**	whole numbers	0	5, 530, -10000
double **(primitive)**	decimal numbers	0.0	5.123, 53.00000, -0.00001
boolean **(primitive)**	true or false	false	true, false
char **(primitive)**	any single character in single quotation marks	the null character	‘5’, ‘e’, ‘*’
String **(reference/object)**	any characters in double quotation marks	null	“hello world”, “5”, “true”

Definitions: object state

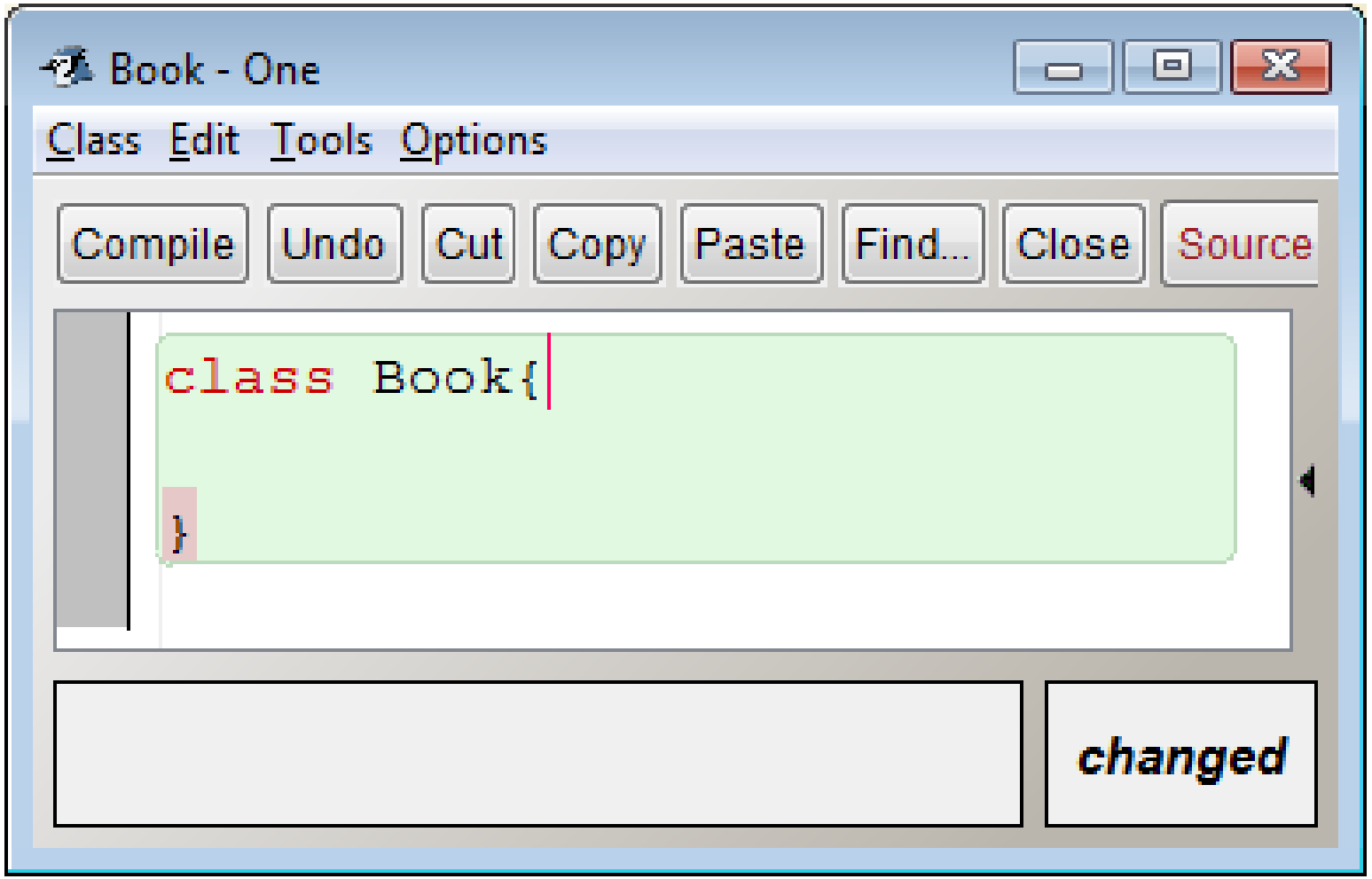
- The set of all values for an object's fields is called the object's state
- These two objects have different states:
- Book 1: title = "Harry Potter" written in 1995 by "J.K. Rowling"
- Book 2: title = "Harry Potter" written in 1996 by "J.K. Rowling"

- These two objects have the same state:
- Book 3: title = "Harry Potter" written in 1995 by "J.K. Rowling"
- Book 4: title = "Harry Potter" written in 1995 by "J.K. Rowling"

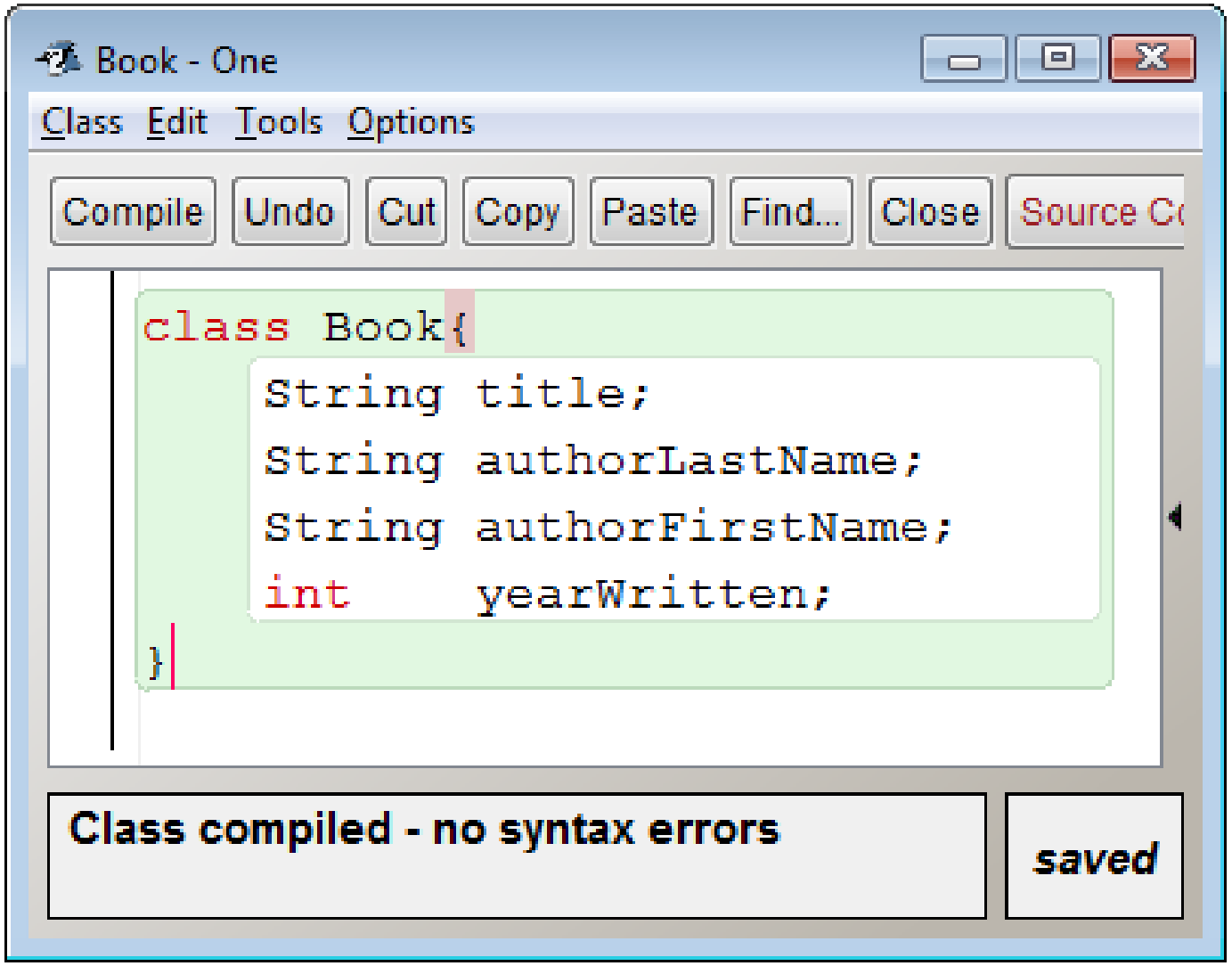
Definitions: methods

- In addition to instance variables, classes can contain behaviors.
- Behaviors are called methods inside a class.
- A method may be also called a function, procedure, a message, or other names
- Methods should be named as verbs
- BankAccount class methods could include:
  - withdraw()
  - deposit()
  - transfer()
  - getBalanceUSD()
  - changePIN()
  - etc….

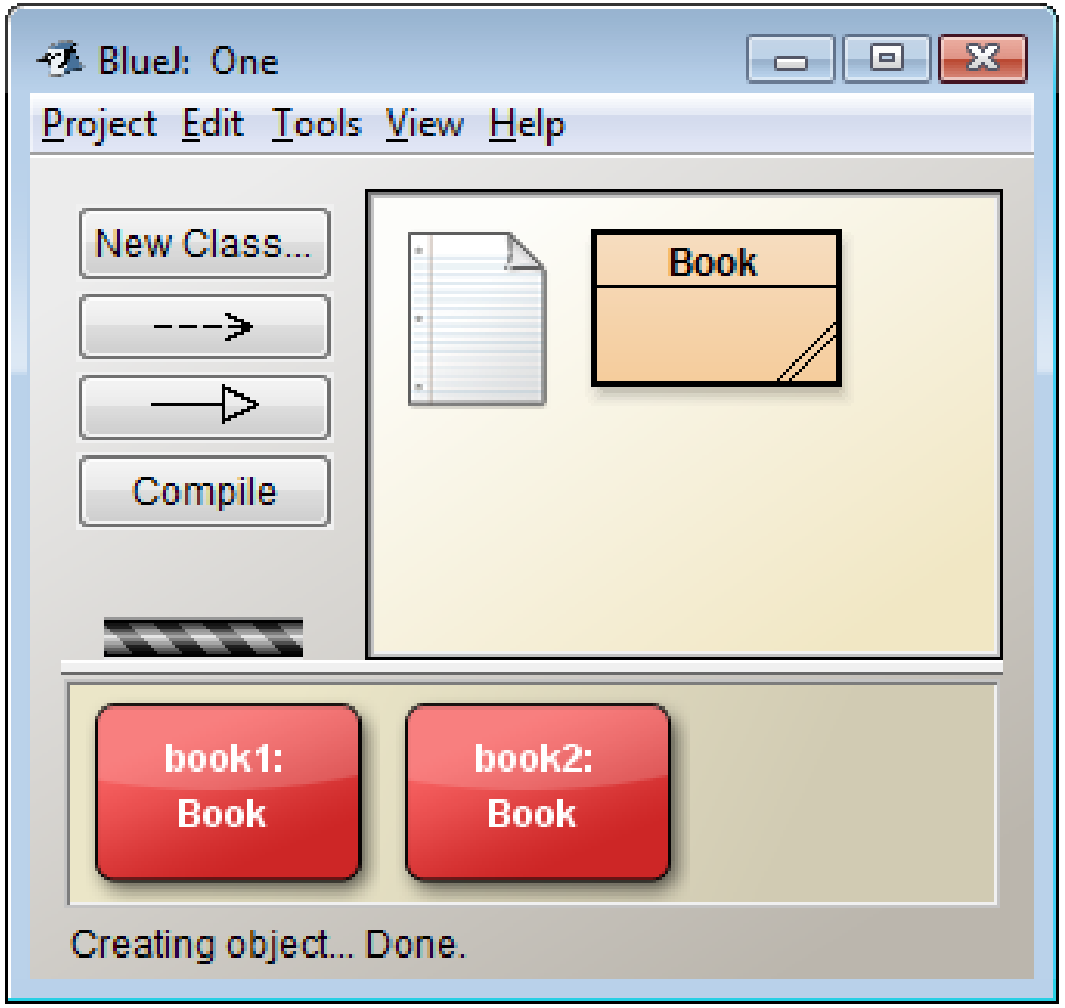
Our first Java class



Our second Java class



Our first Java objects



Our first Java object's state

Lab 1 – In Class

- To be done with a partner in class today.
- Each student must submit their own copy of the lab for me to review and sign off.
- Keep the signed lab until the final day of class to hand them in for marks.

Note: Your work on the lab computers using BlueJ is not saved permanently. It is not important for today's lab, but for take-home labs and assignments you will want to use your own computer.

Next Week

- Quiz on Lesson 1:
  - Definitions
  - Primitive Data Types
  - Naming Conventions
  - Simple Java Class
- More fundamentals of the Java language and classes

Please make sure to do the following before next class:

- Make sure you can access COMP 1409 on learn.bcit.ca
- Install BlueJ on your own computer