

COMP6700/2140 Course Review

Alexei B Khorev and Josh Milthorpe

Research School of Computer Science, ANU

26 May 2017

Course Marks

- The total mark is calculated from continuous assessment and exam marks:

$$T = H_t + AM_1 + AM_2 + \max(0.2 * FE, MSE) + FE$$

- final exam (50):

$$(Q_1, Q_2, Q_3, Q_4) = (10, 15, 10, 15)$$

$$FE = 0.5 * Q_1 + \max(0.5 * Q_1, Quiz) + Q_2 + Q_3 + Q_4$$

- mid-semester exam (10)
- assignments (30):

$$AM_1 = \max(A_1, Q_2) \text{ if } (A_1 \geq 4.5); A_1 \text{ otherwise}$$

$$AM_2 = \max(A_2, Q_4) \text{ if } (A_2 \geq 6); A_2 \text{ otherwise}$$

- homework exercises, H_1, H_2, \dots (10):

$$H_t = \min\left(\sum_{i=1}^8 H_i, 10\right)$$

If your T is greater than 49 (after integer rounding), you will pass the course, if it is $45 \leq T \leq 49$ you will be permitted a supplementary examination.

Course Themes

Java Basics

Object-Oriented Programming

Algorithms and Data Structures

Functional Programming

Graphical User Interfaces

Programming Practice and Tools

Exam Questions

Q1. Java Basics (10)

Q2. Object-Oriented Programming (15)

Q3. Algorithms and Data Structures (10)

**Q4. Functional Programming
and
Graphical User Interfaces (15)**

(Programming Practice and Tools)

Java Basics

Data representation & types: primitive vs. reference, arrays, String

Arithmetic: integer and float

Program structure: expressions, statements, blocks, methods, parameters

Program control: loops, conditionals, methods, recursion

Exceptions

String formatting

I/O: read & write to console, files

Packages and modules

Object-Oriented Programming

Objects vs. classes

Fields, constructors, methods

Extending, overriding

Interfaces and abstract classes

Object equality

Enums

Parameterized (generic) types, generic methods, wildcards

Algorithms and Data Structures

Abstract data types vs. data Structures

Iteration

List, Queue, Set, Map (+Java Collections Framework)

Algorithmic complexity (efficiency)

Recursive Algorithms

Sorting and searching

Functional Programming

Lambda expression, method references

Anonymous inner classes

Functional interfaces

Stream data processing: intermediate and terminal operations

Map, reduce, filter, collect

Graphical User Interfaces

Event-driven programming: handlers, filtering

Scene creation, layout

Programmatic vs. declarative style

Transforms & animations

How Should I Study for the Exam?

- Sample final exam (to be released)
- Homework exercises
- Lab exercises
- Assignments (see [assignment 1 sample solution](#))
- Lecture slides
- More practice: <http://codingbat.com/java>