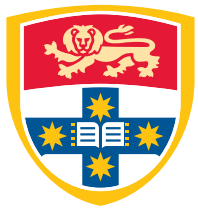
INFO1103: Introduction to Programming

School of Information Technologies, University of Sydney



Week 13: Revision

We will cover: Summarise concepts covered in this course, Examination preparation

You should read: Your perfected notes first! and every textbook reference given

Lecture 25: Revision

Revisiting the main topics

Essentials about a program

What is a program?

What is a compiler?

What is syntax?

What does it mean for a compilation error or syntax error?

None of these answers should involve Java

Variables

How is information represented, i.e. what are the different datatypes?

Identify which data type is most appropriate for the data being represented

Understanding:

- Declaring a variable
- Setting a variable value
- Reading a variable value
- Operations on a variable value

Explain what is a String

Control flow: if statements and loop

What are boolean expressions?

What are the truth tables for AND/OR/NOT

Given a boolean expression, calculate the outcome as either true/false

Identify which path of code is being executed based on given data (flowchart)

Construct a loop to iterate in a very specific manner by controlling when the loop terminates

Tracing variables inside a loop with a deskcheck

Array

Explain what is an array

- Declaring an array variable
- Setting an array element value
- Reading an array element value
- Operations on an array element value

What could be different for each of the above for an *array variable* when compared to a single primitive type variable?

Testing

Why do we bother to test programs if *it works* for the first time?

How many tests are needed before the program is deemed reliable?

Is there such thing as a good test?

Explain what is unit testing

Create tests for a given problem description and code

Create tests for a given problem description *without the code*

What are Exceptions, and how are they different to errors

Idioms, standard stuff

Quite valuable skills to measure if you have passed the course

Know how to write code to sum an array of numbers[1]

Write code to search an array based on a specific condition (based on the current element value)

Write code to search an array based on tracking one thing (e.g. find minimum)

Write code to search an array based on tracking two things (e.g. find index of minimum)

Write a method to take input parameters, do some processing and return a result (not printing!)

[1] otherwise fail

Methods (static)

What is a method and what are the features of a method prototype?

What good are methods if the entire program can be written in the main() method?

What happens to the variables and calculations in the method when it finishes?

How does a method deal with errors that it cannot handle?

Describe what happens to the values of the arguments when calling a method `foo(3, someObject);`
What happens when reading or writing the value of a parameter inside a method?

Classes

What is a class?

We can write a full working program in main() method without additional classes. *When* do we need classes?

What is encapsulation?

What is a [default] constructor?

What are instance variables?

How are instance variables accessed?

Objects

What is an object?

How do you create an Object?

How to use the constructor for an object

What is the different when between using the static and non-static methods?

Does calling a method on an object change data stored in memory? if so, which data?

Methods (instance)

Write a method to allow read only access to private instance variables

as above, but for a reference type (hint copy)

Write a method to initialise all instance variables as default (e.g. multiple constructors)

Write a method to evaluate the class invariant

Design a method that would change the state of an object, instance variables, based on a give description

Files: Input/Output

How can data be represented in a file?

How do you read/write each character of of a file at a time?

How do you read/write each line of of a file at a time?

How do you read the entire contents of a file into working memory?

Recursion

A method `R` calls another method with the same signature

...method `R`, who calls another method with the same signature

...method `R`, who calls another method with the same signature

...method `R`, who calls another method with the same signature

...method `R` which terminates on a base case.

What is a base case?

Given recursive code:

- Identify what is the base case in the code?
- Identify the useful work being done in every recursive step
- What is the output of the recursive method, calculate what is returned/printed to screen

Inheritance

If we can create a new class for each type, why do we care about inheritance?

Identifying the correct behaviour of a child/parent class when calling an *overloaded* method on for an object

Compare and contrast abstract class and interface

Creating a new class based on an interface