SWEN20003
Object Oriented Software Development
Workshop 2

Workshop

This week, we are introducing the fundamental piece of abstraction used in Java: *classes*. These exercises will take you through the process of defining a class, including its *attributes* and *methods*. Remember:

- A **class** is a "blueprint" setting out the data associated with a type of object (**attributes**), and the actions the object can perform (**methods**).
- A **object** is an **instance** of a class, containing its own data (separate from other objects of that class).
- A **method** operates on a particular object's data.
- **Static attributes** are shared between all objects of a given class.
- **Static methods** are not associated with any particular object.

1. Create a `Circle` class with a *radius*, *x coordinate*, and *y coordinate*.
   (a) Add a default constructor `public Circle()` that sets the radius to `1` and the coordinates to `(0, 0)`.
   (b) Add a constructor `public Circle(double radius)` that sets the radius to the argument value, and the coordinates to `(0, 0)`.
   (c) Add a constructor `public Circle(double radius, double x, double y)` with the appropriate actions.
   (d) Add `toString` and `equals` methods.

2. Create a similar `Rectangle` class with a *left coordinate*, a *top coordinate*, a *width*, and a *height*.

3. (a) Create a `Book` class to represent a book in a library. Books have an *author*, a *title*, and can either be *borrowed* or *not borrowed*.
   (b) Write a constructor for your class.
   (c) Define getters for your class.
   (d) Add appropriate `toString` and `equals` methods to your class.
   (e) Define a method `void borrow(String borrowedBy)` that marks the book as *borrowed*. You'll need to add an attribute to the class to store who has borrowed the book.
   (f) Define a method `void returnBook()` that returns the book to the library.
   (g) Add a static attribute to count the number of books that are currently borrowed.
   (h) Define a static method that returns the number of books currently borrowed.

4. (a) Create a `Library` class with an appropriate constructor to represent a library that can hold up to `10` books. (Hint: use an array!)
   (b) Define a method to add a book to the library. If the library is already full, it should do nothing.

(c) Define a method `Book lookup(String title)` that looks up a book by title and returns the first book with that title in the library. If there is no such book, it should return `null`.

(d) Add an overloaded method `Book lookup(String title, String author)` that looks up a book by title *and* author.

(e) Add a method `String getCatalogue()` that returns a string containing each book in the library on separate lines, in the following format:

```
Charles Dickens: Great Expectations
Sun Tzu: The Art of War
Brian Kernighan & Denis Ritchie: The C Programming Language
```

(Hint: if you define `Book`'s `toString` method carefully, this problem is easy.)

(f) Replace the static attribute and method in the `Book` class with an instance variable and method in the `Library` class.

(g) Write a `main` method to create some books, add them to a library, look up books, and borrow them.