

CMPSC 100

Computational Expression

ArrayLists

- An `ArrayList` is essentially its namesake: a list of collected `objects`.
 - I place an emphasis here on the word *objects*: an `ArrayList` can only collect `reference types` and output of `wrapper classes` (`objects`)
 - This includes:
 - String objects
 - Integer, Double objects (`wrapper classes` only; not `primitives`)
 - Any object created by a reference type, such as:
 - Our `BankAccount` objects from previous labs/practicals
 - Our `Bill` objects from previous labs/practicals
 - Our `Fraction` objects from previous labs/practicals

ArrayLists

Creating a “copy” of a class

- The “syntax” (or way you write/create these **objects**) may seem a little wonky:

```
ArrayList<E> {IDENTIFIER} = new ArrayList<E>();
```

Reference/wrapper type

```
ArrayList<Fraction> fractions = new ArrayList<Fraction>();
```

```
ArrayList<String> names = new ArrayList<String>();
```

ArrayLists

- `ArrayLists` store objects in spaces with numeric indexes:

0: "Ulysses"

1: "The Boss"

2: "Snooze Magoo"

- And we can "get" or reference these by their numeric address.

ArrayLists

```
ArrayList<String> catNames = new ArrayList<String>();  
catNames.add("Ulysses");  
catNames.add("The Boss");  
catNames.add("Snooze Magoo");  
System.out.println(catNames.get(1));
```

> The Boss

- See page 230 in *JSS* for a more detailed list.

Exercise #2

Navigate to the `ArrayList` folder in our `class-activities/23-october` folder.

Exercise #2

The guidelines:

- Take user input of:
 - Title
 - Author
- Add input to ArrayList

```
package arraylist;

public class Book {
    public final String title;
    public final String author;

    public Book (String title, String author) {
        this.title = title;
        this.author = author;
    }
}
```



```
String response = new String();  
String title, author;  
Book book;  
ArrayList<Book> library = new ArrayList<Book>();
```

```
while (!response.equalsIgnoreCase("N")) {  
    System.out.print("Enter book title: ");  
    title = input.nextLine();  
    System.out.print("Enter book author: ");  
    author = input.nextLine();  
    book = new Book(title, author);  
    library.add(book);  
    System.out.print("Add another book? [Y/N]: ");  
    response = input.nextLine();  
}
```

```
System.out.println("Our library contains " + library.size() + " books:");
    int index = 0;
    while (index < library.size()) {
        book = library.get(index);
        System.out.println(book.title + " by " + book.author);
        index++;
    }
```

DISPLAYBOOKS.JAVA

```
String response = new String();
String title, author;
Book book;
ArrayList<Book> library = new ArrayList<Book>();
while (!response.equalsIgnoreCase("N")) {
    System.out.print("Enter book title: ");
    title = input.nextLine();
    System.out.print("Enter book author: ");
    author = input.nextLine();
    book = new Book(title, author);
    library.add(book);
    System.out.print("Add another book? [Y/N]: ");
    response = input.nextLine();
}
System.out.println("Our library contains " + library.size() + " books:");
int index = 0;
while (index < library.size()) {
    book = library.get(index);
    System.out.println(book.title + " by " + book.author);
    index++;
}
```

DISPLAYBOOKS.JAVA

Exercise #2

Test using `gradle -q --console plain run`