

15-100 Final Exam Information

Fall 2007

The final examination for **all sections** of 15-100 will be given on **Monday, December 17** from **8:30-11:30AM**. Your instructor will tell you what room to report to for your exam.

Topics

The final examination this semester will be a written exam that will be based on the following topics from 15-100:

- Primitive data types, type-casting
- Arithmetic, relational and logical operators
- The `String` class in Java
- Conditional statements (`if`, `if-else`)
- Loops (`while`, `do-while`, `for`)
- One-dimensional arrays
- Fundamental Algorithms: finding the min or max of an array, finding the position of the min or max in an array, finding the average of data stored in an array, counting the number of items in the array that match some criteria, shifting/moving data in an array
- Object-oriented programming: using a previously-defined class; creating a class to model a real-world object by writing constructors, mutators/setters, and accessors/getters; writing `equals` and `toString` methods
- Maintaining a collection of object references using an array: inserting a new reference, removing a reference, traversing a collection, creating a new collection of references from an existing collection that meet some specific criteria
- Recognizing when these exceptions occur: array index out of bounds, string index out of bounds, trying to use a null reference to invoke a method
- Reading an API for a class in order to use its methods correctly

Students are **not** allowed to use any additional material (books, notes, etc.) during the exam. Students will be supplied with an API for the `String` class on the exam. (See next page.)

On the day of the final exam

- **Arrive early. Students arriving late will not be given extra time.** Please arrive 15 minutes before the exam period begins to get to your assigned seat so you can hear the exam instructions completely.
- **Missing the exam will result in a failing grade for the exam.** If you are ill or have a family emergency during your exam time, you must have a valid documented excuse and must speak with your instructor as soon as possible.
- **Bring 2 pens or pencils and your student ID card** (or some other government-issued photo ID) with you. Your ID will be checked at the exam. If you do not bring an ID, you may be asked to leave the exam room.
- **You may not use electronic devices of any kind during the exam.** This includes, but is not limited to, cell phones, pagers, calculators, PDAs, computers, music players, etc. It is better if you do not even bring these items to the exam room in the first place.
- **Please use the restroom before arriving at your exam room.**

```
public final class String
extends Object
implements Comparable
```

sample **String** API

The **String** class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Constructor Summary

```
String(String original)
```

Initializes a newly created **String** object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.

Method Summary

```
char charAt(int index)
```

Returns the character at the specified index.

```
int compareTo(String anotherString)
```

Returns a negative integer, zero, or a positive integer if this string is lexicographically less than, equal to, or greater than the string specified in the argument, respectively.

```
boolean equals(String anotherString)
```

Returns true if this string has the exact same sequence of characters as the specified string. Otherwise, returns false.

```
int length()
```

Returns the length of this string.

```
String replace(char oldChar, char newChar)
```

Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.

```
String substring(int beginIndex)
```

Returns a new substring of this string from beginIndex to the end of this string.

```
String substring(int beginIndex, int endIndex)
```

Returns a new substring of this string from beginIndex up to but not including endIndex.

```
String toLowerCase()
```

Returns a new string that is a copy of this **String** converted to lower case.

```
String toString()
```

This object (which is already a string!) is itself returned.

```
String toUpperCase()
```

Returns a new string that is a copy of this **String** converted to upper case.