# CS102A Introduction to Computer Programming

# Fall 2020

# Lab 8

## Objectives

1. Learn to declare constructors and use them to construct objects.
2. Learn to use various String methods.

## 1 Prework: Constructors and Instance Methods

The Circle class defined in Lab 7 does not contain any explicitly declared constructors. As such, the Java compiler will provide a default constructor initializing all three fields (radius, x, y) to 0.0 when invoked. If we want to create a Circle object whose three fields are initialized according to the following values: radius = 2.0, x = 1.0, y = 1.0, we can rewrite our main method as follows:

```
public static void main(String[] args) {
    Circle c = new Circle();
    c.setRadius(2.0);
    c.setX(1.0);
    c.setY(1.0);
}
```

However, this is quite troublesome. A better, more elegant solution is to declare custom constructors that can be used to construct Circle objects with specific values for radius, x, and y.

The following code declares two such constructors. The first constructor takes one argument to initialize the `radius` field (`x` and `y` are initialized to `0.0`). The second constructor takes three arguments to initialize all three fields:

```java
public Circle(double radius) {
    this.radius = radius;
}

public Circle(double radius, double x, double y) {
    this.radius = radius;
    this.x = x;
    this.y = y;
}
```

Note that, in the constructors, the keyword `this` is needed to differentiate between constructor arguments and fields belonging to the class. Now, we can simply create a `Circle` object (`radius = 2.0`, `x = 1.0`, `y = 1.0`) with a simple constructor call.

```java
public static void main(String[] args) {
    Circle c = new Circle(2.0, 1.0, 1.0);
}
```

Type the following code in the `main` method and see what happens:

```java
Circle c = new Circle();
```

The above code would not compile. If it not clear to you why this happen, please check the lecture notes.

# 2 Exercises

## 2.1 Exercise 1

Add a `public` method `distanceToOrigin()` to the `Circle` class. This method should return the distance between the circle's center point and the origin point `(0.0, 0.0)`. Then, write a Java program `Lab8E1.java` that performs the following tasks:

1. Generate a random integer $N$ in the range $[5, 10)$.

2. Create $N$ circles. Each circle has a random radius in the range $[1.0, 3.0)$ and a random center position: `x` and `y` are in the range $[2.0, 5.0)$.

3. Among the generated circles, find the one with the smallest area and the one whose center is the farthest from the origin point.

**Tip**

For random number generation, you may use the following two methods of the `Random` class:

```
public int nextInt(int bound)
```

```
public double nextDouble()
```

See `https://docs.oracle.com/javase/8/docs/api/java/util/Random.html` for more details.

Sample input and output:

```
Circle #1: radius = 1.71, x = 4.84, y = 4.46
Circle #2: radius = 2.90, x = 2.78, y = 4.08
Circle #3: radius = 2.63, x = 4.29, y = 2.63
Circle #4: radius = 1.24, x = 2.17, y = 4.85
Circle #5: radius = 2.14, x = 3.00, y = 4.21
Circle #6: radius = 2.92, x = 4.36, y = 2.39
Circle #7: radius = 2.55, x = 4.57, y = 4.87
Circle #8: radius = 2.59, x = 2.14, y = 2.12
Circle #9: radius = 2.02, x = 3.40, y = 3.04
Circle #4 is the smallest circle, area = 4.87
Circle #7 is the farthest circle, distance to origin = 6.68
```

## 2.2 Exercise 2

**Tip**

Please use `String` methods (`https://docs.oracle.com/javase/8/docs/api/java/lang/String.html`) to finish the tasks below. Methods in the `Character` class are also helpful: `https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html`.

Write a program `Lab8E2.java` that checks whether a string provided by a user is a palindrome or not. A string is a palindrome if the reverse of the string is the same as the original string (we do not differentiate between upper- and lower-case characters in this exercise). For example, `abba`, `#Aa#`, and `0` are palindromes. Your program should continuously read and assess user inputs, stopping when the user types `quit`.

Sample input and output:

```
Type a string ("quit" to finish): hello
hello is not a palindrome
Type a string ("quit" to finish): many
many is not a palindrome
Type a string ("quit" to finish): 0
0 is a palindrome
Type a string ("quit" to finish): 900
900 is not a palindrome
Type a string ("quit" to finish): #Aa#
#Aa# is a palindrome
Type a string ("quit" to finish): quit
```

## 2.3 Exercise 3

Write a program `Lab8E3.java` that removes all repeated characters in a string provided by the user, returning a new string without any repeated characters or white spaces. You may use `StringBuilder` to build the new string.

Sample input and output:

```
Please type a string: hello
After removing repeated chars: helo
```

```
Please type a string:
Empty string, exiting...
```

```
Please type a string: abcd  bcde cdef
After removing repeated chars: abcdef
```

## 2.4 Exercise 4

Write a program `Lab8E4.java` that counts the occurrences of substring `s2` in string `s1`. Specifically, the program should ask the user to input two strings `s1` and `s2`, and output the following:

1. The index of `s1` where each occurence of `s2` starts; and
2. The total number of occurrences of `s2` in `s1`.

Sample input and output:

```
s1: JavaExamplesJavaCodeJavaProgram
s2: Java
Found at index: 0
Found at index: 12
Found at index: 20
Total occurrences: 3
```

```
s1: abcd  bcde cdef
s2: bc
Found at index: 1
Found at index: 6
Total occurrences: 2
```

```
s1: abcdefg
s2: xyz
Total occurrences: 0
```

## API References

String methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

- public int length()
- public char charAt(int index)
- public boolean startsWith(String prefix)
- public boolean equals(Object anObject)
- public boolean equalsIgnoreCase(String anotherString)
- public String trim()
- public int indexOf(String str)
- public int indexOf(String str, int fromIndex)
- public String substring(int beginIndex)
- public String substring(int beginIndex, int endIndex)
- public String[] split(String regex)
- public char[] toCharArray()

Character methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html

- public static char toLowerCase(char ch)
- public static boolean isWhitespace(char ch)

StringBuilder methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html

- public StringBuilder append(char c)
- public String toString()