

## Contents

Introduction	1
Reintroduction to Java	1
Data Structures	4

## Introduction

Data structures is the study of how to organize information in a computer so as to ensure efficiency. Note that I am not taking this class purely of my own volition, so I will be much more sarcastic in these notes than even the PDEs notes.

## Reintroduction to Java

Everyone here has learned how to write code in Java,<sup>1</sup> so we're going to go over a quick review of everything we learned in Java.

The variable types are as follows:

- String: text, like "Hello";
- int: integers, like 123;
- double: floating point numbers, like 19.99;
- char: characters, like 'a';
- boolean: stores the states true or false.

---

### Hello World

```
1 public class Main{
2     public static void main(String[] args) {
3         System.out.println("Hello, World.");
4     }
5 }
```

---

Note that unlike Python, we need to specify the data type of each variable. For instance,

---

### Values to Variables

```
1 String message;
2 message = "Hello, World";
3
4 int value1;
5 value1 = 15;
6
7 double value2;
8 value2 = 24.8;
```

---

To obtain values from user inputs, we need to use the Scanner library.

---

### User Input

```
1 import java.util.Scanner;
2 public class Main{
3     public static void main(String[] args){
```

---

<sup>1</sup>Well, "learned" is a strong word.

```

4     Scanner input = new Scanner(System.in);
5
6     System.out.println("Integer:");
7     int a=input.nextInt();
8
9     System.out.println("Double:");
10    double b = input.nextDouble();
11
12    System.out.println("Text:");
13    input.nextLine(); //Need this, else will return blank line
14    String c=input.nextLine();
15 }
16 }

```

We can also include if/else statements.

#### Using If/Else Statements

```

1 public class Main{
2     public static void main(String[] args){
3         int a=10;
4         int b=2;
5         if(a > b){
6             System.out.println("a is greater than b");
7         } else if (a < b){
8             System.out.println("b is greater than a");
9         } else {
10            System.out.println("a and b are equal");
11        }
12    }
13 }

```

The loop syntaxes<sup>II</sup> are as follows:

#### While Loop

```

1 int a=0;
2 int b=0;
3 int c=5;
4
5 while (a < c){
6     b = b + 10;
7     a = a + 1;
8 }

```

#### For Loop

```

1 int a;
2 int b=0;
3 int c = 5;
4 for(a=0; a < c; a= a + 1){
5     b = b+10;
6 }

```

The next most important structure we use a lot is the Array/Array List.

#### Arrays and Array List

```

1 import java.util.*;
2 public class PlayingWithArrays{
3     public static void main(String[] args){
4         List<Integer> a = new ArrayList<>();
5         a.add(10);
6         a.add(11);
7         a.add(12);
8         System.out.println(a);

```

---

<sup>II</sup>Syntices?

```

9      a.set(0,20);
10     System.out.println(a.get(0));
11
12     int[] b = {30,31,32};
13     System.out.println(b[0]);
14     System.out.println(Arrays.toString(b));
15 }
16 }

```

---

Note that array lists are data structures, as well as arrays.

Java also admits functions (but, in classic Java fashion, they are called methods).

### Functions and Methods

---

```

1 public class Main{
2     public static double area (int base, int height){
3         double result;
4         result = base * height/2;
5         return result;
6     }
7 }

```

---

Java is an object-oriented language, so there are all the fun parts of OOP, like classes, instances, etc.

### Classes and Instances

---

```

1 class Professor{
2     String first_name;
3     String last_name;
4     String email_address;
5     String office_location;
6 } // A class.
7 public class Main{
8     public static void main(String[] args){
9         // Instances
10        Professor the_reader = new Professor();
11        Professor not_the_reader = new Professor();
12    }
13 }

```

---

The two most important types of methods in Java are the getter and the setter.

### Getting and Setting

---

```

1 class Movie{
2     private String title;
3     public void setTitle(String title){
4         this.title = title; //use of this keyword tells us that we want to change the title that is
5         part of the class Movie rather than the title that is our argument.
6     } // our title setter
7     public String getTitle(){
8         return this.title; //Similarly, this returns the value of our previously set title.
9     }
10 }
11 public class Main{
12     public static void main(String[] args){
13         Movie our_american_programming_language = new Movie();
14         our_american_programming_language.setTitle("Our American Programming Language"); //Use setter
15         to set title
16         System.out.println(our_american_programming_language.getTitle()); //Use getter to access
17         title (without changing it)
18     }
19 }

```

---

## Data Structures

We use data structures to achieve efficiency, in the sense that it uses the least time, fewest instructions, and least memory usage. When evaluating an algorithm, we need to understand all of these methods.

Consider a guessing game where one participant chooses a number and the other participant tries to guess. There are two ways to guess:

- guessing directly (i.e., just guessing in increasing order);
- guessing via binary search.

The latter approach is significantly more efficient than the former.