Contents

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,[1] 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

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

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

Values to Variables

```
String message;
message = "Hello, World";

int value1;
value1 = 15;

double value2;
value2 = 24.8;
```

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

User Input

```
import java.util.Scanner;
public class Main{
  public static void main(String[] args){
    Scanner input = new Scanner(System.in);

    System.out.println("Integer:");
```

[1] Well, "learned" is a strong word.

```
    int a=input.nextInt();

    System.out.println("Double:");
    double b = input.nextDouble();

    System.out.println("Text:");
    input.nextLine(); //Need this, else will return blank line
    String c=input.nextLine();
  }
}
```

We can also include if/else statements.

Using If/Else Statements

```
  public class Main{
    public static void main(String[] args){
      int a=10;
      int b=2;
      if(a > b){
        System.out.println("a is greater than b");
      } else if (a < b){
        System.out.println("b is greater than a");
      } else {
        System.out.println("a and b are equal");
      }
    }
  }
```

The loop syntaxes[II] are as follows:

While Loop

```
    int a=0;
    int b=0;
    int c=5;

    while (a < c){
      b = b + 10;
      a = a + 1;
    }
```

For Loop

```
  int a;
  int b=0;
  int c = 5;
  for(a=0; a < c; a= a + 1){
    b = b+10;
  }
```

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

Arrays and Array List

```
import java.util.*;
public class PlayingWithArrays{
  public static void main(String[] args){
    List<Integer> a = new ArrayList<>();
    a.add(10);
    a.add(11);
    a.add(12);
    System.out.println(a);
    a.set(0,20);
    System.out.println(a.get(0));

```

[II] Syntices?

```
    int[] b = {30,31,32};
    System.out.println(b[0]);
    System.out.println(Arrays.toString(b));
  }
}
```

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

```
public class Main{
  public static double area (int base, int height){
    double result;
    result = base * height/2;
    return result;
  }
}
```

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

Classes and Instances

```
class Professor{
  String first_name;
  String last_name;
  String email_address;
  String office_location;
} // A class.
public class Main{
  public static void main(String[] args){
    // Instances
    Professor the_reader = new Professor();
    Professor not_the_reader = new Professor();
  }
}
```

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

Getting and Setting

```
class Movie{
  private String title;
  public void setTitle(String title){
    this.title = title; //use of this keyword tells us that we want to change the title that is
        part of the class Movie rather than the title that is our argument.
  } // our title setter
  public String getTitle(){
    return this.title; //Similarly, this returns the value of our previously set title.
  }
}
public class Main{
  public static void main(String[] args){
    Movie our_american_programming_language = new Movie();
    our_american_programming_language.setTitle("Our American Programming Language"); //Use setter
         to set title
    System.out.println(our_american_programming_language.getTitle()); //Use getter to access
        title (without changing it)
  }
}
```