Things you should know

1. Variables and Method implementation in Java can only be inside a Class.

2. The structure of a class is

```
public class MyClass {

}
```

In the code above, public is the access type, which indicates the class is publicly accessible. class is the name of the structure we use in Java, which is the template or definition of an object. MyClass is the name of the class. Any variables or methods should only be inside one such structure.

3. A class with public access has to be defined in file with the same name. The above class has to be stored as MyClass.java else the code will not compile.

4. The class can have static and non-static attributes and methods. Static attributes and methods are common for all the objects of the class we create. Non-static attributes and methods can only be accessed through an Object of the class. The attributes can have different values for each Object of the class.

5. A method in Java is defined like this.

```
<access_type> [static] <return type> <method name>(<parameters>) {
}

public static int addNumbers(int i, int j) {
return i+j;
}
```

The above is a static method which returns an int. It takes two parameters of type int. Let's assume this method is in a class called Calculator. To call this methods we should say,

```
Calculator.add(5,6);
```

6. Every statement in Java should end with a semi-colon. Class and Method implementation will be followed by code block which starts with { and ends with }. 7. To create an object of a class we use,

```
  = new
//This will create a new String object
String str = new String();

//This will create a new Customer object
Customer customer = new Customer();
```

8. A class can have more than on constructor. Constructor is a special method without return type which constructs the object. The parameters to the constructor depends on how the class has defined it. 9. Most classes have getters and setters for all the attributes, which are public. The access of the methods can be changed by the programmer as per design requirements. 10. There are two kinds of collection objects in Java. They are static collection and dynamic collection. 11. Static collection is the one where the size of the collection is defined and cannot be changed.

```
//creates an int array of size 10.
int[] iArr = new int[10];
//creates a String array of size 15.
String[] strArr = new String[15];
```

12. Dynamic collection can vary in size dynamically. In any dynamic collection, when the collection is created, the coder should predefine the kind of objects it is going to hold. =>ArrayList is one of the most common collection. ArrayList can have any valid objects. It can also contain null. ArrayList allows duplication.

```
//Creates an arraylist which can hold String objects
ArrayList arrListString = new ArrayList<>();
```

=>HashSet is a collection which cannot have duplicates. But it can have null.

```
//Creates an hashset which can hold Integer objects
HashSet hashsetInt = new HashSet<>();
```

=>TreeSet is a collection which cannot have duplicates. It can't have null. It can only have objects which have Comparator implemented because the TreeSet sorts the objects that are stored in it. If you use a TreeSet to store Strings, they are automatically sorted in alphabetic order.

```
//Creates an treeset which can hold String objects
TreeSet treeSet = new TreeSet<>();
```

=> HashMap is collection which is represented as key-value pairs. The Key and the Value have to be valid objects. The keys are unique. The objects can be duplicated.

```
//Creates a hash map of String to Customer.
HashMap myHashMap = new HashMap<>();
```

=> TreeMap is collection which is represented as key-value pairs. The Key and the Value have to be valid objects. The keys are unique. The objects can be duplicated. The keys are sorted in order irrespective of the order it is put inside the TreeMap. The class which the Key is made of should have Comparator interface implemented.

```
//Creates a tree map of String to Customer.
TreeMap myHashMap = new TreeMap<>();
```