# Steps in Running Program.

System Design > July9 > MOM.txt

```
    interface vs abstract class
-------------------------------------------------------------
Object creation
    1. Space allocation & this assignment
        Space is allocated for all non-static data members. this is
        set equal to the new object's address
        Data members are assigned default values (JAVA)
            0 = byte, short, int, long
            0.0 = float, double
            false = boolean
            '\0' = char
            null = everything else
    2. Parsing
        Class creator's default values are assigned to data members
        (but only such values which are non-dynamic)
        e.g. growth multiple is set to 1.5 in ArrayList
        e.g. rehashing threshold is set in HashMap to 0.5
    3. Constructor
        3.1 Default = Class creator's default values are assigned to
        data members (but only such values which are dynamic such as
        student folder or dbconnection)
        3.2 Parametrized = For class user to set values which he
        requires

        Notes -> 2:54 to 3:04
        Doubts -> 3:04 to 3:14

```

System Design > July9 > f1.txt

```
public static void main(String[] args) throws
NumberFormatException, IOException {
  Student s = new Student(10, "A");
}

static class Student {
  // During space allocation all data members are set to JAVA
  given default values
  // Parsing sets data members to Class creator's defaults (e.g.
  maxxScore and schoolName here and growthMultiple in ArrayList)
  int age;
  String name;
  String house;
  String schoolName = "DAV";
  int maxScore = 100;
  DBConnection dbconnection;
  Directory studentFolder

  // For class creator to initialize data members which are
  dynamic e.g. reading from a file or a db
  Student(){
      dbconnection = new DBConnection("<ip address>");
      studentFolder = Files.read("..");
  }

  // For class user to set values to data members
  Student(int age, String name){
      this.age = age;
      this.name = name;
  }
```

# Final Variable / Functions:

Datamembers once denoted final can be assigned only once, then cannot be reassigned.

if forgot to set value in parsing, then we can assign its first value in static block or constructor.

```
static class Person {
  final int maxScore = 100;
  void fun(){
    final int age = 10;
    //age++; → error
    System.out.println("Max score is = " + maxScore);
  }
}
```

error

maxScore++; ↑

final methods / functions cannot be overriden. (cant be written again in inherited classes).

final methods can be overloaded, (its signature is changed even if same function).