Building Distributed Data Applications

In [ ]:
```
1. Define set of types (classes, objects, case classes) required to suppo
offers multiple courses and each course can be multiple enrollments. A st
than one course at a time. Create type of events (like Enrollment, Course
enrollment, cancel a class etc.) and simulate handling those events as th
mutable state in each data type.
a. Must use case classes
b. Must use pattern matching to process various events
c. Must have an object to provide the entry point to the application to s
d. All events must be of a common type
e. May use for or other list functions to print various courses offered a
time, when an information request is presented

case class School
case class Course
case class Enrollment
case class Student
//create events - enrollments, course, cancel enrollment, cancel class
```

In [2]:
```
var a = Array("")
```

```
a: Array[String] = Array("")
```

In [3]:
```
a
```

```
res2: Array[String] = Array("")
```

In [4]:
```
import scala.collection.mutable.ArrayBuffer
var a = new ArrayBuffer[String]
```

```
import scala.collection.mutable.ArrayBuffer
a: collection.mutable.ArrayBuffer[String] = ArrayBuffer()
```

In [5]:
```
import scala.io.Source._
val lines = scala.io.Source.fromFile("/Users/APARNA/Downloads/Project - D
```

```
java.io.FileNotFoundException: /Users/APARNA/Downloads/Project - Devel
oper - apache-access-log (No such file or directory) (/Users/APARNA/Do
wnloads/Project - Developer - apache-access-log (No such file or direc
tory))
  java.io.FileInputStream.open(Native Method)
  java.io.FileInputStream.<init>(FileInputStream.java:146)
  scala.io.Source$.fromFile(Source.scala:91)
  scala.io.Source$.fromFile(Source.scala:76)
  scala.io.Source$.fromFile(Source.scala:54)
  cmd4$$user$$anonfun$1.apply(Main.scala:24)
  cmd4$$user$$anonfun$1.apply(Main.scala:23)
```

In [ ]:
```

```