

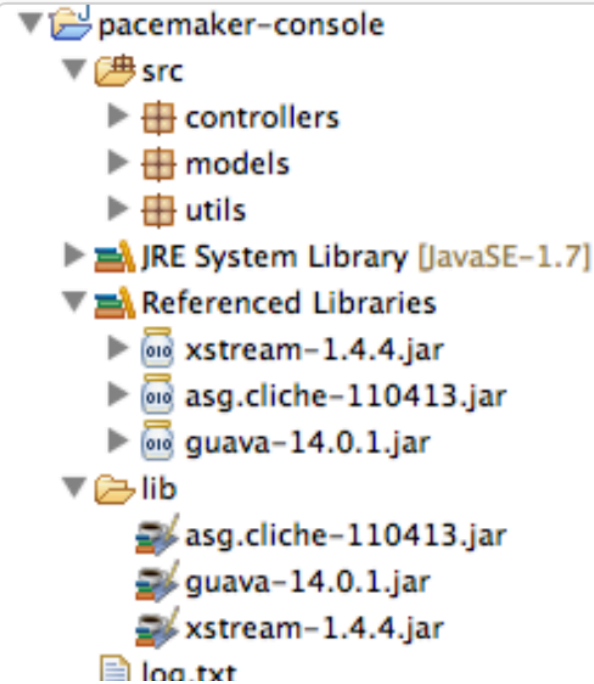
---

## Lab-01 Eclipse & Java



Variables	
Name	Value
args	String[0] (id=15)
users	ArrayList<E> (id=16)
elementData	Object[10] (id=29)
[0]	User (id=32)
email	"bart@simpson.com" (id=35)
firstName	"Bart" (id=38)
lastName	"Simpson" (id=39)
password	"secret" (id=41)
[1]	User (id=43)
email	"bart@simpson.com" (id=35)
firstName	"Homer" (id=44)
lastName	"Simpson" (id=39)
password	"secret" (id=41)
modCount	2
size	2

## Lab-02 CLI & Classes



## Lab-03 Objects & Serialization



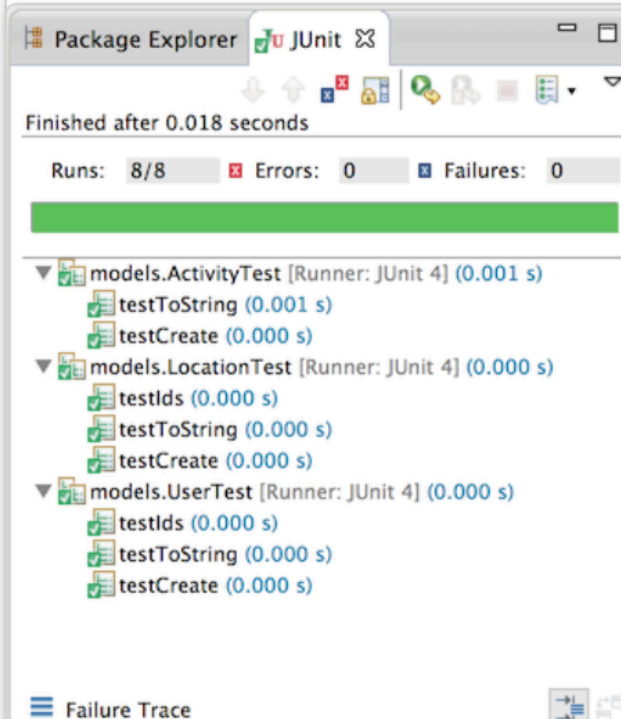
Node	Content
object-stream	
map	
entry	
entry	
entry	
entry	
entry	
map	
entry	
string	homer@simpson.com
models.User	
id	0
firstName	homer
lastName	simpson
password	secret
activities	
email	homer@simpson.com
entry	
map	
entry	
long	0
models.User	
id	0
firstName	homer
lastName	simpson
email	homer@simpson.com
password	secret
activities	
entry	

Extend the pacemaker application

## Lab-04 Testing



The objective of this lab is to set up your development workstation and to bring in some of the terms and concepts covered in the opening lab.



## Lab-05 Refactoring



The layout of a variant 2 (Leach-Salz) UUID is as follows:

```
UUID (Java Platform SE 8 )
0xFFFFFFFF00000000 time_low
0x00000000FFFFFFFF time_mid
0x0000000000000000 version
0x0000000000000000 time_hi
```

The least significant long consists of the following unsigned values:

```
0xC000000000000000 variant
0x3FFF000000000000 clock_seq
0x0000FFFFFFFFFFFF node
```

Refactor pacemaker to employ uuid instead of long ids. Unsure if the tests are still passing as we make this transition. Make a start command line formatting features.

## Lab-06 Maven

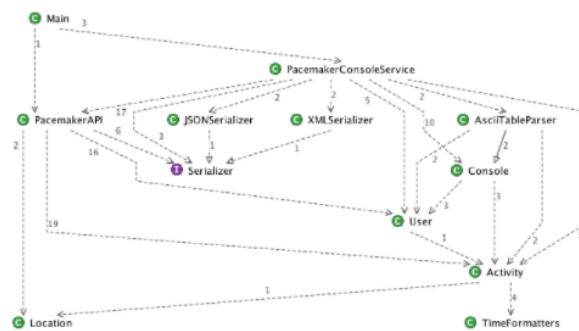


```
$ mvn -version
Apache Maven 3.1.0 (893ca28a1da9d5f51ac03827af98bb7
Maven home: /Users/edeleastar/dev/apache-maven-3.1.
Java version: 1.7.0_40, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.7
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.8.5", arch: "x86_
$
```

In the previous lab, you installed Maven. In this lab, we will incorporate Maven into our pacemaker-console-lab05 solution. We will also use Maven to bring JUnit5 capabilities into Eclipse.

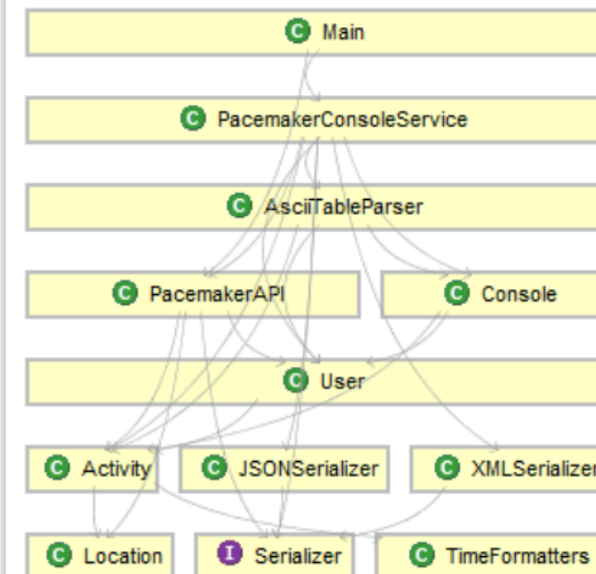
Standard	Core Features [30%]	Presentation [20%]	Tests [30%]	Build Systems [20%]
Baseline	Users/Activities/ Locations (lius, la, du)	Plain	basic API tests	none
Good	Start DateTime (la sortBy:) Persistence - XML (l, s)	Pretty	full API tests	maven (build)
Excellent	Persistence -JSON (cff)	Tabular	UI Tests	maven (test)
Outstanding	Persistence - YAML OR Extra Reports	Enhanced	accurate coverage report submitted	maven (modular approach)

## Lab-07a Pacemaker Models



Build a sample solution to Assignment 1, using TDD techniques, Maven & Eclipse

## Lab-07b Pacemaker API



Complete the pacemaker assignment solution.

## Lab-08 Skeleton



```
app.get("/users", ctx -> {
    service.listUsers(ctx);
});

app.post("/users", ctx -> {
    service.createUser(ctx);
});

app.get("/users/:id", ctx -> {
    service.listUser(ctx);
});

app.get("/users/:id/activities", ctx -> {
    service.getActivities(ctx);
});
```

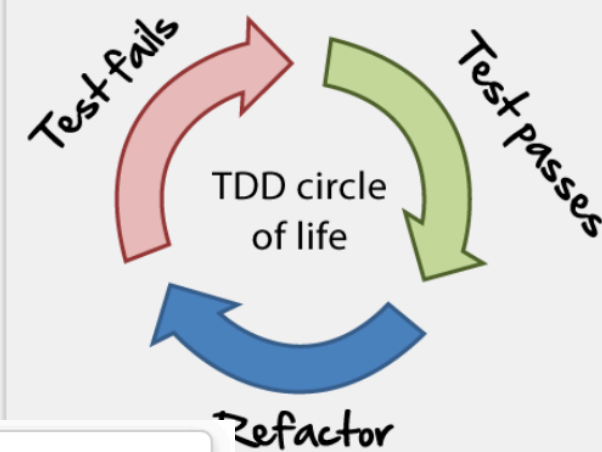
Develop a baseline for Assignment 2, to include a simplified version of pacemaker application developed so far

## Lab-09 Simple Rest API



Evolve a simple the existing pac app using mmicrofi

## Lab-10 Rest CLI + Test



test client  
so, build a  
r the servi

## Lab-11 Kotlin Rest Service



Rewrite aspects of the Pacemaker Skeleton Service in Kotlin. Verify that translation via the Java test and CLI clients.

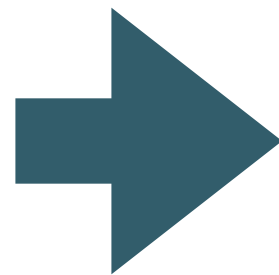
## Lab-12 Kotlin Rest CLI + test



Rewrite the Java Test and CLI clients in Kotlin.

# Assignment : Pacemaker 2.0

Create a new version  
of Pacemaker,  
evolved to explore 4  
lines of inquiry



- Commands/Features
- Test Driven Development Practices
- Build & Deployment
- Language Features

## Commands/Features (1/4)

List Users: List all users emails, first and last names	gu get-users ()
Register: Create an account for a new user	ru register-user (first name, last name, email, password)
Login: Log in a registered user in to pacemaker	lu login-user (email, password)
Logout: Logout current user	l logout ()
Add activity: create and add an activity for the logged in user	aa add-activity (type, location, distance)
List Activities: List all activities for logged in user	la list-activities ()

## Commands/Features (2/4)

Add location: Append location to an activity"	al add-location (activity-id, lat, lng)
List Activity Location: List all locations for a specific activity	lal list-activity-locations (activity-id)
ActivityReport: List all activities for logged in user, sorted alphabetically by type	ar activity-report ()
Follow Friend: Follow a specific friend	f follow (email)
List Friends: List all of the friends of the logged in user	lf list-friends ()
Friend Activity Report: List all activities of specific friend, sorted alphabetically by typer	far friend-activity-report (email)



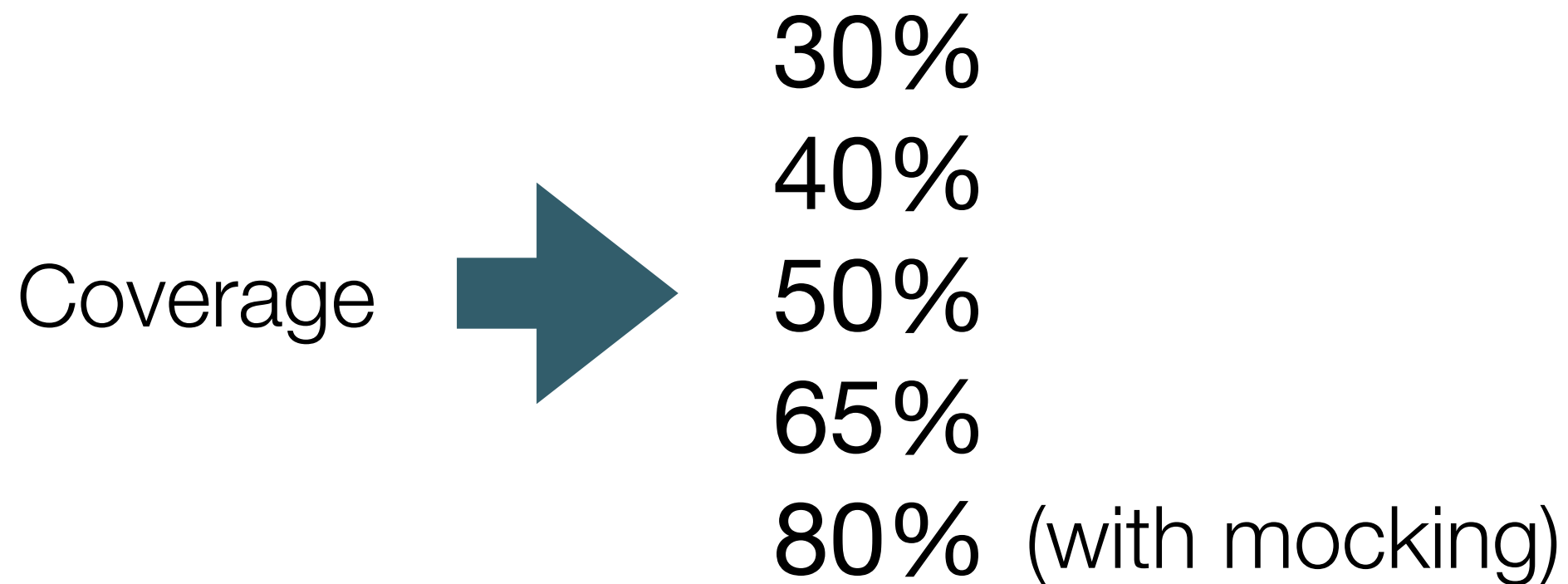
## Commands/Features (3/4)

Activity Report: List all activities for logged in user by type. Sorted longest to shortest distance	ar activity-report (byType: type)
Unfollow Friends: Stop following a friend	uf unfollow-friend ()
Message Friend: send a message to a friend	mf message-friend (email, message)
List Messages: List all messages for the logged in user	lm list-messages ()
Distance Leader Board: list summary distances of all friends, sorted longest to shortest	dlb distance-leader-board ()
Friend Activity Report: List all activities of specific friend, sorted alphabetically by type	ar activity-report (byType: type)

## Commands/Features (4/4)

Distance Leader Board: distance leader board refined by type	dlbdt distance-leader-board-by-type (byType: type)
Message All Friends: send a message to all friends"	maf message-all-friends (message)
Location Leader Board: list sorted summary distances of all friends in named location	llb location-leader-board (location)

# Test Driven Development Practices



## Build & Deployment

Eclipse project archive

- pacemaker-console

---

github repo

- pacemaker-console

---

maven github repos:

- pacemaker-service
- pacemaker-console

---

pacemaker-service provides REST API

pacemaker-console access API (over http)

---

pacemaker-service deployed to cloud

pacemaker-client access cloud service

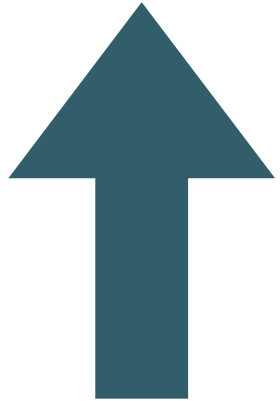


simple

useful



# Language Features



verbose

Java

Java with Lambdas

Java with Streams OR Kotlin

Kotlin



concise

# Grading Spectrum

Grade Band	Packaging & Deployment	Commands	TDD Coverage	Language
Starter	Eclipse project archive - pacemaker-console	gu get-users () ru register-user (first name, last name, email, password) lu login-user (email, password) l logout () aa add-activity (type, location, ..	30%	Java
Baseline	github repo - pacemaker-console	al add-location (activity-id, longitude, latitude) lal list-activity-locations (activity-id) ar activity-report () f follow (email) lf list-friends () far friend-activity-report (email)	40%	Java
Good	maven github repos: - pacemaker-service - pacemaker-console	ar activity-report (byType: type) uf unfollow-friend () mf message-friend (email, message) lm list-messages () dlb distance-leader-board ()	50%	Java with Lambdas
Excellent	pacemaker-service provides REST API pacemaker-console access API (over http)	dlbbt distance-leader-board-by-type (byType: type) maf message-all-friends (message) llb location-leader-board (location)	65%	Java with Streams OR Kotlin
Outstanding	pacemaker-service deployed to cloud pacemaker-client access cloud service	Admin Account  Define commands to administer service, to include: - remove users - disable/enable users - report user stats (nmr logins, average number of activities etc...)	80% With Mocking	Kotlin

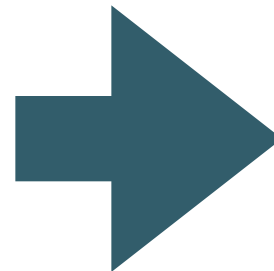
# Lab 06

## Lab-06 Maven



```
$ mvn -version
Apache Maven 3.1.0 (893ca28a1da9d5f51ac03827af98bb7
Maven home: /Users/edelestar/dev/apache-maven-3.1.
Java version: 1.7.0_40, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.7
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.8.5", arch: "x86_
$
```

In the previous lab, you installed Maven. In this lab, we will incorporate Maven into our pacemaker-console-lab05 solution. We will also use Maven to bring JUnit5 capabilities into Eclipse.



Guidance on implementing  
maven modules

### Packaging & Deployment

Eclipse project archive  
- pacemaker-console

github repo  
- pacemaker-console

maven github repos:  
- pacemaker-service  
- pacemaker-console

pacemaker-service provides  
REST API  
pacemaker-console access  
API (over http)

# Lab 08

## Lab-08 Skeleton



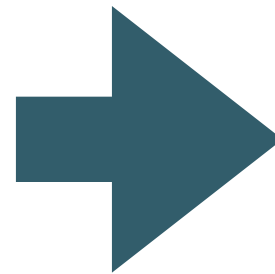
```
app.get("/users", ctx -> {
  service.listUsers(ctx);
});

app.post("/users", ctx -> {
  service.createUser(ctx);
});

app.get("/users/:id", ctx -> {
  service.listUser(ctx);
});

app.get("/users/:id/activities", ctx -> {
  service.getActivities(ctx);
});
```

Develop a baseline for Assignment 2, to include a simplified version of pacemaker application developed so far



Provide foundation  
application structure +  
implement starter  
commands

### Commands

```
gu get-users ()
ru register-user (first name, last name,
email, password)
lu login-user (email, password)
l logout ()
aa add-activity (type, location,
...
)

al add-location (activity-id, longitude,
latitude)
lal list-activity-locations (activity-id)
ar activity-report ()
f follow (email)
lf list-friends ()
far friend-activity-report (email)

ar activity-report (byType: type)
uf unfollow-friend ()
mf message-friend (email, message)
lm list-messages ()
dlb distance-leader-board ()

dlbbt distance-leader-board-by-type
(byType: type)
maf message-all-friends (message)
llb location-leader-board (location)
```



Lab-09 Simple Rest API

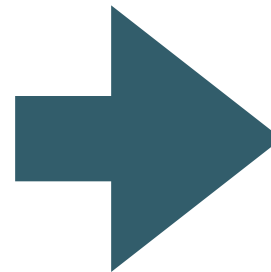


  
**REST API**

Evolve a simple Rest service from the existing pacemaker-skeleton app using the Javalin mmicroframework.


## Lab 09

Evolve Starter commands  
into REST service



Packaging & Deployment
Eclipse project archive - pacemaker-console
github repo - pacemaker-console
maven github repos: - pacemaker-service - pacemaker-console
pacemaker-service provides REST API pacemaker-console access API (over http)

Lab-10 Rest CLI +  
Test



Test fails

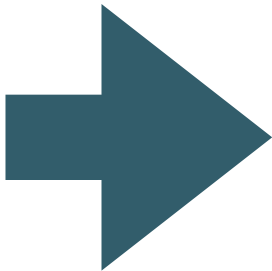
Test passes

Refactor

TDD circle of life

Compose a test client for the REST services. Also, build a minimal CLI for the service.

Lab 10



Evolve tests for APIs

TDD Coverage
30%
40%
50%
65%
80% With Mocking

### Lab-11 Kotlin Rest Service



Rewrite aspects of the Pacemaker Skeleton Service in Kotlin. Verify that translation via the Java test and CLI clients.

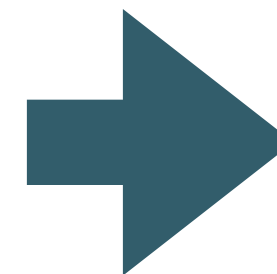
## Lab 11 & 12

### Lab-12 Kotlin Rest CLI + test



Rewrite the Java Test and CLI clients in Kotlin.

Explore Kotlin  
implementations of  
pacemaker starter service



Language
Java
Java
Java with Lambdas
Java with Streams OR Kotlin
Kotlin