# 1 Experimental Description

The next part of this document is concerned with what will be evaluated and how. The following sections will define the methodology and approach that will be taken.

## 1.1 Tasks to complete

The participants will be given three main tasks to complete with various subtasks.

1. Iterator example in Mungo. Code provided. Spot and correct the error. Add the remove operation.
2. SMTP[1] example in Scribble and Mungo. Code provided. StMungo introduced as well. Spot and correct the error one in scribble specification, second one in Mungo specification.
3. Implement a small example based on an informal specification. Buyer seller would be a good candidate here.

## 1.2 Variable/Experimental conditions

Independent variables:

- Programming expertise

Dependent variables to be measured:

- time to complete each task
- how many times the code is compiled
- how many bugs/errors
- is the end result correct

## 1.3 Data Collection/Measurements/Observations

The purpose of this study is exploratory. Dependent variables to be measured:

- time to complete each task
- how many times the code is compiled
- how many bugs/errors are encountered in the process
- is the end result correct
- participants' opinion about the tools/session types

## 1.4 Design

Considering that the aim of this experiment is exploratory and the learning effect (performance improving with experience, even over very short periods of time) is of little concern in this case a within subjects design, every participant perform every task under every condition, will be used.

[1] http://www.ietf.org/rfc/rfc2821.txt

## 1.5 Participants/Subjects/Users (Controlled Variable)

The only requirement for participants is that they have some Java experience. Factors like level of programming proficiency, background, or upfront knowledge of behavioural types or session types will be taken into consideration.

## 1.6 Experimental location

The experiment will be carried out in Sir Alwyn Williams building, room F112. This office has been chosen as it allows a higher degree of control over the environment, the programming one in particular.

## 1.7 Experimental schedule

This section outlines all of the parts of the experiment and a rough estimate how long each will take. After completing all tasks, anticipated to take roughly 45 minutes, the participants will be asked to answer a short survey rating their experience using session types. The time taken to complete the experiment may be longer or shorter depending on the level of programming expertise.

1. 5 minutes to explain the experiment
2. 15 minute practical tutorial on session types, typestate, the tools to be used
3. approximately 10 minutes for task 1
4. approximately 15 minutes for task 2
5. approximately 20 minutes for task 3
6. 5 minute short informal interview
7. 5 minute to complete survey
8. 5 minutes for any questions the participant might have