CLAIMS - Experimenter Documentation

I. Pre-experiment

Before actual experimentation the experimenter must execute four short utility programs which set up files for the CLAIMS experiment. To do this:

```
    )CLEAR

    )FLOAD SET UP_CLAIMS
```

The first program to be executed is:

```
    SET UP_EXPERIMENT
```

This program literally starts you from ground-zero. It destroys all previous pre-experiment files. This program should only be executed once: at the beginning of pre-experiment work.

```
    SET UP_VARIABLES
```

This program must be executed at least once before the CLAIMS experiment can run. The VARIABLES include the maximum allowable loans and short sales; broker commissions, (a percent plus a constant); the interest rate; the length of the trading quarter; the amount of the payoff per claim at the end of each quarter; and the beginning cash and claim balances for the subjects. This program can be executed over again to set and reset VARIABLES if mistakes are made, or adjustments are deemed necessary.

```
    SET UP SUBJECTS
```

This program must be executed at least once before the CLAIMS experiment can run. The subjects´ names are the necessary data for this program. This program assigns each subject to an account number. This program can also be executed over again to correct spelling or to add subjects to the list.

SET UP PAYOFF

This program must be executed at least once for each session the CLAIMS experiment is to be used. Thus, if the experimenter is planning for 1Ø sessions of the CLAIMS experiment, 1Ø PAYOFF files (one for each session) must be set up.

The first half of this program sets up the SPECIAL INFORMATION part of the experiment. (Some subjects are given or are sold information which help them determine which claim will pay off.) A series of questions is asked as to the extent and type of SPECIAL INFROMATION to be used.

The second half of this program requires that you set up a PAYOFF PROBABILITY TABLE. The data for this table is the probable outcome of each claim paying off.

Example:

Quarter	XXX	YYY	ZZZ
1	5Ø	25	25
2	35	35	3Ø

Thus, the probability of YYY paying off in quarter 1 is 25%. The probability of ZZZ paying off in quarter 2 is 3Ø%. (Claims has since been modified to a two stock experiment.)

II. Experiment - the experimenter

To start a session:

```
)CLEAR
)FLOAD CLAIMS
```

then type the name of the driver function:

CLAIMS

If any of the above pre-experiment files have not been set up, CLAIMS will type out a diagnostic and exit the program. CLAIMS will ask for a session number. (CLAIMS will not let you repeat a session in order to protect the possible destroying of data.) CLAIMS will then ask for the number of brokers you want to participate in the session.

The subjects may then begin to log on. CLAIMS monitors the log-on process and makes sure each subject has logged on properly, and has completed their INDIFFERENCE CURVE (explained later).

When the log-on process is completed, CLAIMS goes into the SPECIAL INFORMATION routine and notifies the experimenter. When all the SPECIAL INFORMATION decisions are in, CLAIMS is ready to begin the trading session. When ready, type an ´S´ to begin the SESSION.

At the end of each QUARTER-BREAK, the experimenter has the choice of

(1) beginning a new quarter, or

(2) ending the session.

When the experimenter does choose to end the session, the TAKE-HOME PAY for each subject is typed out on the experimenter´s terminal.

III. Experiment - the subject

ACCOUNT NUMBER: The subject must be given his own personal ACCOUNT NUMBER which he must input into the terminal at the beginning of each session. A successful input of the number will print out the subject´s name and his initial cash and claim holdings at the beginning of the session. The currency, pounds (), is a contrived one; in that 1000 has an actual value of $0.01. Thus, 25,000 is equivalant to $0.25. The actual claim security is valued by the subject himself in the trading market. At the end of each TRADING QUARTER, one of the three claims pay off, the other two do not. All claims pay off the same fixed amount, (which is set in SET UP VARIABLES).

INDIFFERENCE CURVE: This part of the program is designed to ask for three points on each subjects´ INDIFFERENCE CURVE between $0.00 and $10.00, by giving subjects a hypothetical choice between a "sure thing" and a "50-50 gamble".

TRADING QUARTER: The subject will try to maximize his profits through the trading of these claims by:

1) Adjusting his protfolio to maximize his claims payoff.

2\ Trading profitably (buying low, selling high)

QUARTER BREAK: Trading is suspended for a period of time which is called the QUARTER BREAK. Four things happen during the break.

1) The claim which pays off for the quarter is announced.

2) An end of the quarter statement of accounts is typed out. (eg. 25,000   250 XXX   250 YYY   250 ZZZ)

3) The TAKE-HOME account, (the amount of money in dollars each subject will be paid for his participation in that session), is typed out. Each quarter the subject "takes home" his cash account, (converted to dollars), plus the amount of his payoff

from his claims.

Example:		
	PREVIOUS BALANCE	$Ø.ØØ
	+ CASH ACCOUNT	Ø.25
	+ PAYOFF OF XXX	Ø.25
	NEW BALANCE	$Ø.5Ø

4) The SPECIAL INFORMATION routine is then executed, and some subjects receive the benifit of new information.

END OF SESSION: At the end of each session the subject is provided with a statement of how much money (TAKE-HOME) he made during that session.

IV. Post-experiment - Raw Data Files

CLAIMS Files: Data is collected on a separate file for each session. All files are labelled:

CLAIMS<session-number>

Thus, CLAIMS1 would be the data collected during session 1. Similiarly for CLAIMS2, CLAIMS3, etc.

Data analysis: To interpret the data on the disk file, note first that each object on the file is an APL vector or matrix.

A matrix object is the internal representation of a stock queue, and is printed out each time the stock queue is altered. The matricies are provided for checking against other data, and provide little information unto themselves. These queues of unmatched orders are preceeded by a two-letter matrix definition:

XB - XXX BIDQUEUE	XO - XXX OFFERQUEUE
YB - YYY BIDQUEUE	YO - YYY OFFERQUEUE
ZB - ZZZ BIDQUEUE	ZO - ZZZ OFFERQUEUE

The queue matrix itself is arranged horizontally from best price to worst price. Vertically, there are five columns. Each represents:

(PRICE) (SHARES) (AN) (BOOK) (UON)

The vector objects are more complicated, and describe specific actions by the subjects at a specific time. Each vector object has a identifying character, describing it. Examine the tables below.

IC	Identification
A	Subjects ACCOUNTS at the end of the quarter break
B	Subject makes a BID for a claim
C	Subject CANCELs an order
D	DONE SESSION, session is completed
E	END OF QUARTER
F	Subjects quarterly SPECIAL INFORMATION decisions
H	Subjects TAKE-HOME PAY for the session
I	Subjects INDIFFERENCE CURVE
O	Subject makes an OFFER for a claim
P	Subject moves to another TRADING POST
Q	QUARTER begins
R	PAYOFF RESULTS during quarter break
S	SESSION begins
T	TRANSACTION between subjects
U	USER or Subject logs on
V	VARIABLES set in SET_UP_VARIABLES
W	SPECIAL INFORMATION variables set in SET_UP_PAYOFF

In each of these logged objects:

OBJECT[1]=Real time in seconds

OBJECT[2]=Subjects Account Number

OBJECT[3]=the identifying character

This is where the similarity of the objects end.

Examine the following table.

OBJECT[3]=A, [4]=Cash, [5]=XXX claims, [6]=YYY claims,
[7]=take home pay

OBJECT[3]=B, [4]=POST, [5]=PRICE, [6]=SHARES, [7]=AN, [8]=BOOK,
[9]=UON

OBJECT[3]=C, [4]=POST, [5]=UON, [6]=PRICE, [7]=SHARES, [8]=AN, [9]=BOOK
[10]=UON

OBJECT[3]=D, [4]=Session Number

OBJECT[3]=E, [4]=Quarter Number

OBJECT[3]=F, [4]=Broker or Special Information Broker,
[5]=Samples or Probabilities,
[6]=Given, Auction, Flat Rate, None, [7]=Price of Information,
[8-12]=Information Samples or [8-9]=Information Probabilities

OBJECT[3]=H, [4]=AN, [5-25]=NAME, [26] TAKE-HOME PAY

OBJECT[3]=I, [4]=POINT at .25, [5]=POINT at .5,
[6]=POINT at .75

OBJECT[3]=O, [4]=POST, [5]=PRICE, [6]=SHARES, [7]=AN, [8]=BOOK,
[9]=UON

OBJECT[3]=P, [4]=POST, [5]=POST going too

OBJECT[3]=Q, [4]=Quarter number

OBJECT[3]=R, [4]=Quarter number, [5]=% payoff X,[6]=% payoff Y
[7]=% payoff Z,[8]=payoff stock, [9]=PAYOFF

OBJECT[3]=S, [4]=Session Number, [5]=Year, [6]=Month, [7]=Day

```
   [8]=Hour, [9]=minute, [1Ø]=Second

OBJECT[3]=T, [4]=POST, [5]=Bought or Sold, [6]=PRICE

   [7]=SHARES, [8]=AN, [9]=BOOK, [1Ø]=UON

OBJECT[3]=U, [4]=Station number, [5]=Broker or Special Information Brok

   [6]=Beginning Cash, [7]=Beginning Shares, [8-28] NAME

OBJECT[3]=V, [4]=Loans, [5]=Short Sales, [6]=% Commission,

   [7]=Constant Commission, [8]=Interest Rate, [9]=Quarter Length

   [1Ø]=PAYOFF, [11]=Beginning Cash, [12]=Beginning Shares

OBJECT[3]=W, [4]=Number of Brokers eligible to receive Special Informat

   [5]=Maximum number of Brokers receiving Special Information each qua

   [6]=Initial Offering Price of Information,

   [7]=Lowest Offering Price of Information,

   [8]=Type of Information: Samples or Probabilities
```

PRICE -- Price in pounds ()
SHARES -- Shares of CLAIMS
AN -- Subject´s Account Number
BOOK -- Ø=Market Order, 1=Book Order
UON -- Unique Order Number, to trace a particular order
POST -- Ø=No Post, 1=XXX, 2=YYY
NAME -- 2Ø Characters, blank filled

V. Post-experiment - Data Analysis

Ten short data analysis programs have been written for the CLAIMS experiment. To use these programs:

```
)CLEAR
)FLOAD SET_UP_CLAIMS
```

The programs can be executed in any order. Each program will ask for a LOG FILE NAME. Type in CLAIMS1 or CLAIMS2 etc., depending on which session you want to do the analysis on. Then start the paper at the top of a new page and hit the RETURN key when ready for printout. The program names are listed below with a short description of each.

RECONSTRUCTION

Writes the LOG FILE out sequentially, entry by entry, into a readable form. (If you want a copy of the LOG FILE exactly as it is on the disk, use []DUMPLOG, a standard APL function.)

TRANSACTION_STATS

Summary of all transactions by quarter for each stock. Simple aggregate statistics program.

PART_STATS_TRAN

Participation Comparison between Special Information Brokers and Regular Brokers using Transaction statistics.

PART_STATS_BIDOFF

Participation Comparison between Special Information Brokers and Regular Brokers using Bid and Offer statistics.

PART_STATS_PAYOFF

Participation Comparison between Special Information Brokers and Regular Brokers using Payoff and Portfolio Value statistics.

PART STATS PRICE_CHANGE

Participation Comparison between Special Information Brokers and Regular Brokers using Distribution of Price Change Statistics

PART STATS_QUOTE

Participation Comparison between Special Information Brokers and Regular Brokers using the statistic of Real Time each held the Highest Bid or Lowest Offer.

TRANSACTION_GRAPH

Graph of all Transactions by quarter for each stock. Provides comparison between Special Information Brokers and Regular Brokers.

QUOTATION GRAPH

Graph of all Bids and Offers by quarter for each stock. Provides comparison between Special Information Brokers and Regular Brokers.

TRAN_VOLUME_HIST

Histogram of all Transaction Volume using 3Ø second intervals, by quarter for each stock. Provides comparison between Special Information Brokers and Regular Brokers.