

Assessment

Due: 3.00pm Wednesday, 4th April, 2007

Worth 3% of your final mark

There is no Peer review for this assignment

SHARE TRADING - stockMarket.bom.nz

The resources for this assignment can be found at the following URL:

www.cs.auckland.ac.nz/compsci101s1c/assignments/

Aim of the assignment

The goals of this assignment are to get you to practise breaking your code into methods, and to use conditional statements.

This assignment consists of one question. For this assignment you need to submit **THREE** Java source files:

- `A2Application.java`
- `A2Program.java`
- `Keyboard.java`

You also need to submit a text file (`A2.txt`) giving your feedback on the assignment – details of what you should include in this text file are described later in this handout.

Things to note:

All the variables declared in your program **MUST** be **LOCAL** variables i.e. all variables must be declared inside a method.

Before you start your assignment, make sure you **read p.6 of this document**.

You have been provided with the application, `A2Application.java`, a skeleton program file, `A2Program.java`, and with a different version of `Keyboard.java`

(see p.6 of this document where there is an explanation of how to run your program).

Your code MUST be broken down into methods. Each method should perform one task only. For example you could use a method to print the current share prices, to print the user's current share holdings, to print the user's current financial information, to obtain the number of shares the user wants to buy/sell, to obtain the total share value, to randomly add 0, 1 or -1 to the current price of a share, to obtain the initial random share price, to print the final result at the end of the program, etc.

Share trading program - Part 1

(90 marks)

You are required to write an interactive share trading program, i.e. the user can use this program to buy and sell shares. At the beginning of the program the user has a total of \$100 to spend and the user has no shares. There are three possible types of shares which the user is able to buy and sell, shares of type A, B and C.

At the start of the program the price of each of these three types of shares is set to a random value of either \$5, \$6, \$7, \$8 or \$9. Firstly the program should display the current share prices and then the program displays the user holdings in each of the three share types (this is always 0 to start with). For example, in the output below, A shares are worth \$9, B shares \$6 and C shares \$6:

```
Current Share Prices A: $9, B: $6, C: $6
Your current share holdings
    Share A($9): 0 value $0
    Share B($6): 0 value $0
    Share C($6): 0 value $0
```

The user should then be asked how many shares of type A they would like to buy or sell. To buy shares the user enters a positive whole number, to sell shares the user enters a negative whole number, and the user enters 0 if they do not wish to buy or sell. The user is then asked whether they wish to buy/sell shares of type B, and shares of type C:

```
Share A (current value $9). Enter number to buy or sell: 2
Share B (current value $6). Enter number to buy or sell: 1
Share C (current value $6). Enter number to buy or sell: 1
```

Whenever the user buys shares there is a fee of \$1. The fee of \$1 is incurred whenever the user buys at least 1 share of any type. The only time this fee is not incurred is if the user buys 0 shares of all three types. The program displays the fee incurred and then displays a summary of the user's current financial situation, i.e. their total funds, the total value of all their share holdings and the funds they have remaining after paying for both the shares and the fee.

```
$1 fee has been incurred
Total: $99, your total share value: $30, available funds: $69
```

At this stage, the share values undergo an adjustment. The price of each share should either increase by \$1, decrease by \$1 or remain unchanged. The change which occurs to

each share should be random. The program then displays the new value of each type of share and also displays a summary of the user's current financial situation. The summary is displayed even if none of the three share prices has changed.

SHARE VALUE ADJUSTMENT - Current Share Prices A: \$9, B: \$7, C: \$6
Total: \$100, your total share value: \$31, available funds: \$69

The above activity constitutes one round, i.e. time 1. The program repeats in the fashion described above for FIVE rounds, i.e. for time 2, time 3, time 4 and time 5. After FIVE rounds the program stops. The program should also stop if the user's total funds (the total of their share holdings added to their available funds) is not more than \$1, even if the five rounds have not been completed.

Finally, before the program ends, three things are displayed to the user:

- the final user share holdings,
- a summary of the user's current financial situation,
- a message describing the result of the user's trading.

The final message depends on the final state of the user's finances. The message will have one of the following three formats depending on the financial situation of the user at the end of the program. This message should be displayed once before the program ends.

Congratulations! \$5 profit

or

Nothing lost, nothing gained

or

Bad Luck! \$5 loss

The program, `A2Program`, which you should download from the Assignment Two web page, contains one loop which has been defined for you. The `A2Program` skeleton program is shown below:

```
public class A2Program {
    public void start() {
        final int START_AMOUNT = 100;
        int availableFunds = START_AMOUNT;
        int totalCurrentValue = availableFunds;
        int time = 0;

        while(totalCurrentValue >= 1 && time < 5) {
            time++;

        }
    }
}
```

Below is some example output from the program. The user input is shown inside the boxes. For the example shown below I have used the `input1.txt` as the input to the program.

>java A2Application <input1.txt

```

-----
-----
Time 1
Current Share Prices A: $9, B: $5, C: $9
Your current share holdings
    Share A($9): 0 value $0
    Share B($5): 0 value $0
    Share C($9): 0 value $0
Share A (current value $9). Enter number to buy or sell: 
Share B (current value $5). Enter number to buy or sell: 
Share C (current value $9). Enter number to buy or sell: 
$1 fee has been incurred
Total: $99, your total share value: $32, available funds: $67

SHARE VALUE ADJUSTMENT - Current Share Prices A: $10, B: $5, C: $9
Total: $101, your total share value: $34, available funds: $67

-----
-----
Time 2
Current Share Prices A: $10, B: $5, C: $9
Your current share holdings
    Share A($10): 2 value $20
    Share B($5): 1 value $5
    Share C($9): 1 value $9
Share A (current value $10). Enter number to buy or sell: 
Share B (current value $5). Enter number to buy or sell: 
Share C (current value $9). Enter number to buy or sell: 
No fee has been incurred
Total: $101, your total share value: $34, available funds: $67

SHARE VALUE ADJUSTMENT - Current Share Prices A: $11, B: $6, C: $10
Total: $105, your total share value: $38, available funds: $67

-----
-----
Time 3
Current Share Prices A: $11, B: $6, C: $10
Your current share holdings
    Share A($11): 2 value $22
    Share B($6): 1 value $6
    Share C($10): 1 value $10
Share A (current value $11). Enter number to buy or sell: 
Share B (current value $6). Enter number to buy or sell: 
Share C (current value $10). Enter number to buy or sell: 
$1 fee has been incurred
Total: $104, your total share value: $59, available funds: $45

SHARE VALUE ADJUSTMENT - Current Share Prices A: $12, B: $6, C: $11
Total: $109, your total share value: $64, available funds: $45

```

```

-----
-----
Time 4
Current Share Prices A: $12, B: $6, C: $11
Your current share holdings
    Share A($12): 3 value $36
    Share B($6): 1 value $6
    Share C($11): 2 value $22
Share A (current value $12). Enter number to buy or sell: -1
Share B (current value $6). Enter number to buy or sell: 1
Share C (current value $11). Enter number to buy or sell: 1
$1 fee has been incurred
Total: $108, your total share value: $69, available funds: $39

SHARE VALUE ADJUSTMENT - Current Share Prices A: $11, B: $7, C: $12
Total: $111, your total share value: $72, available funds: $39

-----
-----
Time 5
Current Share Prices A: $11, B: $7, C: $12
Your current share holdings
    Share A($11): 2 value $22
    Share B($7): 2 value $14
    Share C($12): 3 value $36
Share A (current value $11). Enter number to buy or sell: 0
Share B (current value $7). Enter number to buy or sell: 1
Share C (current value $12). Enter number to buy or sell: 0
$1 fee has been incurred
Total: $110, your total share value: $79, available funds: $31

SHARE VALUE ADJUSTMENT - Current Share Prices A: $10, B: $8, C: $11
Total: $108, your total share value: $77, available funds: $31

Your current share holdings
    Share A($10): 2 value $20
    Share B($8): 3 value $24
    Share C($11): 3 value $33
Total: $108, your total share value: $77, available funds: $31

Congratulations! $8 profit

```

Note: Each time you run your code the output will be different because the share prices are initially random and are randomly adjusted.

Share trading program - Part 2

(10 marks)

Each time the user asks to buy or sell shares, your program should check:

1. Are there enough funds to pay for the shares?

If there aren't enough funds the program should display the message, "ERROR - not enough funds", and NO shares of that type should appear as having been bought.

2. Are there enough shares?

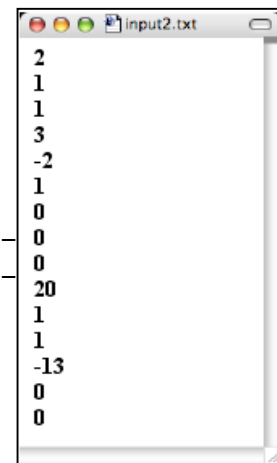
If the user chooses to sell shares (the user enters a negative number) and if there aren't enough shares of that type, the program should display the message, "ERROR - not enough shares", and NO shares of that type should appear as having been sold.

Running your A2Program

In order to help you test your program, A2Program, I have provided TWO input files: input1.txt and input2.txt. I have also provided a different version of the Keyboard class; this version allows you to use file redirection. In order to run your program you can run the application, A2Application, and enter commands from the keyboard or you can run the application with the user input taken from either of the two input files, for example,

```
> java A2Application <input2.txt
```

will execute the application using the user input from the file, input2.txt. The example output and the text file, input2.txt, are shown below:



```
>java A2Application <input2.txt
```

```
-----
Time 1
```

```
Current Share Prices A: $5, B: $9, C: $8
```

```
Your current share holdings
```

```
    Share A($5): 0 value $0
```

```
    Share B($9): 0 value $0
```

```
    Share C($8): 0 value $0
```

```
Share A (current value $5). Enter number to buy or sell: 2
```

```
Share B (current value $9). Enter number to buy or sell: 1
```

```
Share C (current value $8). Enter number to buy or sell: 1
```

```
$1 fee has been incurred
```

```
Total: $99, your total share value: $27, available funds: $72
```

```
SHARE VALUE ADJUSTMENT - Current Share Prices A: $4, B: $8, C: $8
```

```
Total: $96, your total share value: $24, available funds: $72
```

```
-----
Time 2
```

```
Current Share Prices A: $4, B: $8, C: $8
```

```
Your current share holdings
```

```
    Share A($4): 2 value $8
```

```
    Share B($8): 1 value $8
```

```
    Share C($8): 1 value $8
```

```
Share A (current value $4). Enter number to buy or sell: 3
```

```
Share B (current value $8). Enter number to buy or sell: -2
```

```
ERROR - not enough shares
```

```
Share C (current value $8). Enter number to buy or sell: 1
```

```
$1 fee has been incurred
```

```
Total: $95, your total share value: $44, available funds: $51
```

SHARE VALUE ADJUSTMENT - Current Share Prices A: \$4, B: \$7, C: \$8
 Total: \$94, your total share value: \$43, available funds: \$51

 Time 3

Current Share Prices A: \$4, B: \$7, C: \$8

Your current share holdings

Share A(\$4): 5 value \$20

Share B(\$7): 1 value \$7

Share C(\$8): 2 value \$16

Share A (current value \$4). Enter number to buy or sell:

Share B (current value \$7). Enter number to buy or sell:

Share C (current value \$8). Enter number to buy or sell:

No fee has been incurred

Total: \$94, your total share value: \$43, available funds: \$51

SHARE VALUE ADJUSTMENT - Current Share Prices A: \$5, B: \$8, C: \$7
 Total: \$98, your total share value: \$47, available funds: \$51

 Time 4

Current Share Prices A: \$5, B: \$8, C: \$7

Your current share holdings

Share A(\$5): 5 value \$25

Share B(\$8): 1 value \$8

Share C(\$7): 2 value \$14

Share A (current value \$5). Enter number to buy or sell:

ERROR - not enough funds

Share B (current value \$8). Enter number to buy or sell:

Share C (current value \$7). Enter number to buy or sell:

\$1 fee has been incurred

Total: \$97, your total share value: \$62, available funds: \$35

SHARE VALUE ADJUSTMENT - Current Share Prices A: \$6, B: \$7, C: \$7
 Total: \$100, your total share value: \$65, available funds: \$35

 Time 5

Current Share Prices A: \$6, B: \$7, C: \$7

Your current share holdings

Share A(\$6): 5 value \$30

Share B(\$7): 2 value \$14

Share C(\$7): 3 value \$21

Share A (current value \$6). Enter number to buy or sell:

ERROR - not enough shares

Share B (current value \$7). Enter number to buy or sell:

Share C (current value \$7). Enter number to buy or sell:

No fee has been incurred

Total: \$100, your total share value: \$65, available funds: \$35

SHARE VALUE ADJUSTMENT - Current Share Prices A: \$5, B: \$6, C: \$8
 Total: \$96, your total share value: \$61, available funds: \$35

```

Your current share holdings
  Share A($5): 5 value $25
  Share B($6): 2 value $12
  Share C($8): 3 value $24
Total: $96, your total share value: $61, available funds: $35

Bad Luck! $4 loss

```

Note: Each time you run your code the output will be different because the share prices are initially random and are randomly adjusted.

Submitting Files

You should submit the following files for this assignment through the web-based Assignment Dropbox.

```

A2Application.java
A2Program.java
Keyboard.java
A2.txt

```

MAKING MORE THAN ONE SUBMISSION

You can make more than one submission - every submission that you make *replaces* your previous submission. **Only your very latest submission will be marked.**

DO NOT SUBMIT SOMEONE ELSE'S WORK:

- If you submit an assignment you are claiming that you did the work. Do not submit work done by others.
- Do not *under any circumstances* copy anyone else's work – this will be penalized heavily.
- Do not *under any circumstances* give a copy of your work to someone else.
- The Computer Science department uses copy detection tools on the files you submit. If you copy from someone else, or allow someone else to copy from you, this copying will be detected and disciplinary action will be taken.

What you should include in the A2.txt file

You **must** include a text file named A2.txt in your submission. There will be a 5 mark penalty for not doing so.

This text file must contain the following information:

```

Your name
Your login name and ID number
How much time did the assignment take overall?
What areas of the assignment did you find easy?
What areas of the assignment did you find difficult?
Which topics of the course did the assignment most help
  you understand?
Any other comments you would like to make

```


Marking Schedule

Style - A2Program Comment at the top of the class. (2) Good indentation. (3) Good identifier names. (3) Program has been broken down into at least six methods which simplifies the code. (7)		/15
Time 1 -Testing the program The 2 lines of dashes and "Time 1" are displayed. (2) The current share prices are displayed correctly (random \$5, \$6, \$7, \$8 or \$9). (5) The current share prices and the current share holdings (currently 0) are displayed correctly. (5) The user is asked whether they want to buy or sell each type of share correctly. (here the user buys at least one share) (5) The \$1 fee is displayed correctly. (3) The summary of the user's financial situation is displayed correctly. (5)		/25
Time 1 -Testing the program The share prices are adjusted by \$1, \$0 or -\$1. The new prices are displayed correctly. (5) A summary of the user's financial situation is displayed correctly. (5)		/10
Time 2 - 5 -Testing the program The time number changes from 2 to 5. (5) The following 6 things happen for time 2, 3, 4, 5: displays the current share holdings correctly prompts the user if they wish to buy/sell for each share type displays the fee incurred displays the summary of the user's financial situation does the share price adjustment displays the summary of the user's financial situation (10)		/15

Testing the program When the user does not buy any shares the message displayed is "No fee has been incurred". (5) The final message contains the correct value and text. (10) When the user sells shares the values of the shares sold is removed from the user's total share value and added to the user's available funds correctly. (10)		/25
--	--	-----

Part 2 - User errors If there aren't enough funds the program should display the required ERROR message (NO shares of that type should appear as having been bought). (5) If the user chooses to sell shares and there aren't enough shares of that type, the program should display the required ERROR message (NO shares of that type should appear as having been sold). (5)		/10
--	--	-----

Grand Total		/100
-------------	--	------

Penalty

Program uses non-local variables -10marks

A2.txt was not submitted -5 marks