THE UNIVERSITY OF SYDNEY

INFO1103 Assignment 1

Due: 9:00am Monday, 4 April 2016

This assignment is worth 10% of your final assessment

Task description

In this assignment you will develop a coffee bot. Customers can interact with the bot, order coffee and select the number of coffee shots in each cup. The system will provide an order summary and confirm the order with the customer. Payment will be processed using only valid coin and note denominations, with change being refunded in an optimal way that uses the least amount of coins and notes required.

You are encouraged to ask questions on Ed using the assignments category. As with any assignment, make sure that your work is your own, and you do not share your code or solutions with other students.

Working on your assignment

You can work on this assignment on your own computer or the lab machines. It is important that you continually back up your assignment files onto your own machine, external drives, and in the cloud.

You are encouraged to submit your assignment on Ed while you are in the process of completing it. By submitting you will obtain some feedback of your progress on the sample test cases provided.

Academic declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy, and except where specifically acknowledged, the work contained in this assignment or project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the the Academic Dishonesty and Plagiarism in Coursework Policy, can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Information Technologies, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and or communicate a copy of this assignment to a plagiarism checking service or in house computer program, and that a copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.

Implementation details

Write a program in Java that enables your coffee bot to interact with customers, select their order, summarise their order and process transaction using cash and coins, using the correct denominations, i.e. $100.00, $50.00, $20.00, $10.00, $5.00, $2.00, $1.00, $0.50, $0.20, $0.10 and $0.05.

Your program must be contained in one file called `CoffeeBot.java` and produce no errors when compiled using `javac` on the lab machines and Ed. Your program will be run using `java CoffeeBot` with given arguments, reading input from standard input and writing output to standard output.

Important – your program will be marked automatically, so make sure that you follow the assignment specifications carefully. Your program output must match the exact output shown in the examples.

Command line arguments

Your program should take two positive integer command line arguments that specify the supply of coffee cups and shots that will be available. You can assume your program only needs to handle integer arguments. If any given arguments are missing or invalid then output an error message that describes the issue and immediately terminate the program. Several examples are shown below.

There are no command line arguments

```
$ java CoffeeBot
No arguments. System terminating.
```

There are not enough command line arguments

```
$ java CoffeeBot 10
Not enough arguments. System terminating.
```

There are too many command line arguments

```
$ java CoffeeBot 10 20 30
Too many arguments. System terminating.
```

The supply of coffee cups is negative

```
$ java CoffeeBot -10 20
Negative supply of coffee cups. System terminating.
```

The supply of coffee shots is negative

```
$ java CoffeeBot 10 -20
Negative supply of coffee shots. System terminating.
```

The supply of both coffee cups and shots is negative

```
$ java CoffeeBot -10 -20
Negative supply chain. System terminating.
```

Greeting the user

Once the supply of coffee cups and shots has been entered correctly, the coffee bot will then greet the customer and ask for their name so that the coffee bot can optimise the user experience.

```
$ java CoffeeBot 1 2
Hello, what's your name?
```

Read and store the name entered and then ask the customer whether they want to order some coffee.

```
Hello, what's your name? Scott
Would you like to order some coffee, Scott? (y/n)
```

The customer can choose to proceed with the order by entering either *y* or *n* as the response. If the response is different from the possible options then display the following message and ask again.

```
Would you like to order some coffee, Scott? (y/n) ok
Invalid response. Try again.
Would you like to order some coffee, Scott? (y/n)
```

If the user decides to not proceed, display the following message and terminate.

```
Would you like to order some coffee, Scott? (y/n) n
Come back next time, Scott.
```

If the user decides to proceed, display the following message and continue onward.

```
Would you like to order some coffee, Scott? (y/n) y
Great! Let's get started.
```

Order selection

Once the user decides to order then display the banner and available supply as shown below.

```
Would you like to order some coffee, Scott? (y/n) y
Great! Let's get started.

Order selection
---------------

There is 1 coffee cup in stock and each costs $2.00.
There is 1 coffee shot in stock and each costs $1.00.
```

Choosing the number of cups

The customer should then be asked how many cups of coffee they want to order.

```
Order selection
---------------

There is 1 coffee cup in stock and each costs $2.00.
There is 1 coffee shot in stock and each costs $1.00.

How many cups of coffee would you like?
```

If the customer wants zero cups, display the following message and terminate.

```
How many cups of coffee would you like? 0
No cups, no coffee. Goodbye.
```

If the number of cups entered is negative, display the following message and terminate.

```
How many cups of coffee would you like? -100
Does not compute. System terminating.
```

If the customer wants more cups than the stock, display the following message and terminate.

```
How many cups of coffee would you like? 100
Not enough stock. Come back later.
```

Choosing the number of coffee shots in each cup

For each cup, ask the customer how many shots of coffee they want. Amounts zero or more are valid.

```
How many cups of coffee would you like? 2

How many coffee shots in cup 1? 0
How many coffee shots in cup 2? 1
```

If the number of shots entered is negative, display the following message and ask again.

```
How many coffee shots in cup 1? -1
Does not compute. Try again.
How many coffee shots in cup 1?
```

If the customer wants more shots than the stock, display the following message and ask again.

```
How many coffee shots in cup 1? 10
There is only 1 coffee shot left. Try again.
How many coffee shots in cup 1?
```

Order summary

Once the order is selected, display an order summary and ask the customer whether they want to proceed to payment. The price of each coffee cup is $2.00 and the price of each coffee shot is $1.00.

```
How many cups of coffee would you like? 3

How many coffee shots in cup 1? 1
How many coffee shots in cup 2? 2
How many coffee shots in cup 3? 3

Order summary
-------------

Cup 1 has 1 shot and will cost $3.00
Cup 2 has 2 shots and will cost $4.00
Cup 3 has 3 shots and will cost $5.00

3 coffees to purchase.
Purchase price is $12.00
Proceed to payment? (y/n)
```

Payment process

Once the customer proceeds, display the amount to be paid and prompt for payment.

```
Order payment
-------------

$12.00 remains to be paid. Enter coin or note:
```

The payment should be entered in the format $10.00, where the $ sign is followed by a valid coin or note denomination. If the payment is invalid, prompt the customer to enter another payment value.

```
$12.00 remains to be paid. Enter coin or note: 10
Invalid coin or note. Try again.
$12.00 remains to be paid. Enter coin or note:
```

After a valid payment have been entered, check if the payment is enough to pay for the purchase. Continue to prompt the customer to enter payment until the purchase is paid for. For example:

```
$10.00 remains to be paid. Enter coin or note: $10.00
$2.00 remains to be paid. Enter coin or note: $2.00
```

Once the purchase has been paid for proceed to printing the receipt and dispensing change.

Refunding change

If the amount of money paid by the customer equals the purchase price, then display the following.

```
$2.00 remains to be paid. Enter coin or note: $2.00

You gave $2.00
Perfect! No change given.

Thank you, Scott.
See you next time.
```

If there is change to be given, it should be dispensed in optimal coin denominations, i.e. using the minimum possible number of notes and coins. The change should be displayed in the format shown below, outputted in order from the highest to the lowest value denomination.

```
$4.00 remains to be paid. Enter coin or note: $100.00

You gave $100.00
Your change:
1 x $50.00
2 x $20.00
1 x $5.00
1 x $1.00

Thank you, Scott.
See you next time.
```

Examples

```
$ java CoffeeBot 10 20
```

```
Hello, what's your name? Tim
Would you like to order some coffee, Tim? (y/n) n
Come back next time, Tim.
```

```
$ java CoffeeBot 30 40
```

```
Hello, what's your name? Jony
Would you like to order some coffee, Jony? (y/n) y
Great! Let's get started.

Order selection
---------------

There are 30 coffee cups in stock and each costs $2.00.
There are 40 coffee shots in stock and each costs $1.00.

How many cups of coffee would you like? 0
No cups, no coffee. Goodbye.
```

```
$ java CoffeeBot 50 60
```

```
Hello, what's your name? Eddy
Would you like to order some coffee, Eddy? (y/n) y
Great! Let's get started.

Order selection
---------------

There are 50 coffee cups in stock and each costs $2.00.
There are 60 coffee shots in stock and each costs $1.00.

How many cups of coffee would you like? 1

How many coffee shots in cup 1? 10

Order summary
-------------

Cup 1 has 10 shots and will cost $12.00

1 coffee to purchase.
Purchase price is $12.00
Proceed to payment? (y/n) n
Come back next time, Eddy.
```

```
$ java CoffeeBot 70 80
```

```
Hello, what's your name? Phil
Would you like to order some coffee, Phil? (y/n) y
Great! Let's get started.

Order selection
---------------

There are 70 coffee cups in stock and each costs $2.00.
There are 80 coffee shots in stock and each costs $1.00.

How many cups of coffee would you like? 1

How many coffee shots in cup 1? 100
There are only 80 coffee shots left. Try again.
How many coffee shots in cup 1? 10

Order summary
-------------

Cup 1 has 10 shots and will cost $12.00

1 coffee to purchase.
Purchase price is $12.00
Proceed to payment? (y/n) y

Order payment
-------------

$12.00 remains to be paid. Enter coin or note: $100.00

You gave $100.00
Your change:
1 x $50.00
1 x $20.00
1 x $10.00
1 x $5.00
1 x $2.00
1 x $1.00

Thank you, Phil.
See you next time.
```

```
$ java CoffeeBot 90 100
```

```
Hello, what's your name? Craig
Would you like to order some coffee, Craig? (y/n) y
Great! Let's get started.

Order selection
---------------

There are 90 coffee cups in stock and each costs $2.00.
There are 100 coffee shots in stock and each costs $1.00.

How many cups of coffee would you like? 2

How many coffee shots in cup 1? 1
How many coffee shots in cup 2? 5

Order summary
-------------

Cup 1 has 1 shot and will cost $3.00
Cup 2 has 5 shots and will cost $7.00

2 coffees to purchase.
Purchase price is $10.00
Proceed to payment? (y/n) y

Order payment
-------------

$10.00 remains to be paid. Enter coin or note: 10
Invalid coin or note. Try again.
$10.00 remains to be paid. Enter coin or note: 10.00
Invalid coin or note. Try again.
$10.00 remains to be paid. Enter coin or note: $10.00

You gave $10.00
Perfect! No change given.

Thank you, Craig.
See you next time.
```

Writing your own testcases

We have provided you with some sample testcases but these do not test all the functionality described in the assignment. It is important that you thoroughly test your code by writing your own testcases.

You should place all of your test cases in a `tests/` directory. Ensure that each test case has the `.in` input file along with a corresponding `.out` output file. We recommend that the names of your test cases are descriptive so that you know what each is testing, e.g. `no-args.in` and `no-change.in`

Submission details

Your attendance in the week 6 tutorial is required for the manual marking component.

You must submit your code in the assignment page on Ed. To submit, simply place your code into the workspace, then click the run button to check your program works and then click the submit button.

You are encouraged to submit multiple times, but only your last submission will be marked.

Marking

6 **marks** are assigned based on automatic tests for the *correctness* of your program.
This component will use our own hidden test cases that cover every aspect of the specification.

4 **marks** are assigned based on a manual inspection of the *style* and *quality* of your code.

Your program will be marked automatically, so make sure that you carefully follow the assignment specifications. Your program output must match the exact output shown in the examples. You are encouraged to submit your assignment while you are working on it, so you can obtain some feedback.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not properly follow the assignment specification, or your code is unnecessarily or deliberately obfuscated.