Computer Science

CompSci 101 S1 C
– ASSIGNMENT ONE –

ASSESSMENT:

- **Programming due:** **3:00pm, Wednesday 14th March, 2007**
- **Peer review due:** **3:00pm, Monday 19th March, 2007**
- **Worth:** **2% of your final mark**

REFERENCES:

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

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

- or you can click the "`Assignments`" link on the CompSci 101 home page.

AIMS OF THE ASSIGNMENT:

- To practise producing output with `System.out.println()`
- To practise simple arithmetic with the primitive types
- To constructively review the work of your peers

This assignment consists of two questions. For each question you need to submit TWO Java source files (for Question 1: `Q1Application.java` and `Q1Program.java`, and for Question 2: `Q2Application.java` and `Q2Program.java`). You also need to submit the `Keyboard.java` source file and a text file called `A1.txt` giving your feedback on the assignment – details of what you should include in this text file are described later in this handout.

NOTES:

- when you have finished the assignment, you should submit all six files through the web-based Assignment Dropbox (there is a link to this on the Assignments page of the 101 website).
- when the assignment deadline has passed, you will be given access to several random assignments submitted by your peers which you must review using the Peer Assessment System (`https://aropa.ec.auckland.ac.nz`).
- you must complete the peer review for this assignment

Question One (18 marks)

For this question you need to write a program which prints information about the cost of a toll call.

The cost of the call depends on the duration of the call:

each block of twenty minutes costs $1.50,
each block of five minutes costs $0.50,
any single minutes cost $0.12.

The input to the program will be the number of minutes for the call. Your program should display:

the number of twenty minute blocks and their cost,
the number of five minute blocks and their cost,
the number of single minutes and their cost,
as well as the total cost for the call.

Several examples of the program running are shown below (input from the user is shown in bold):

```
> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: 73

Number of minutes: 73
  Twenty minute blocks: 3 ---- $4.5
  Five minute blocks:   2 ---- $1.0
  Single minutes:       3 ---- $0.36
---------------------------------------
Cost of call $5.86

> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: 41

Number of minutes: 41
  Twenty minute blocks: 2 ---- $3.0
  Five minute blocks:   0 ---- $0.0
  Single minutes:       1 ---- $0.12
---------------------------------------
Cost of call $3.12

> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: 3
```

```
Number of minutes: 3
  Twenty minute blocks: 0 ---- $0.0
  Five minute blocks:   0 ---- $0.0
  Single minutes:       3 ---- $0.36
---------------------------------------
Cost of call $0.36
```

Your output must be *identical* to that shown above:

- there must be quotation marks around the words "CALL PRICER"

- You can assume that the user only enters valid input. The number of minutes will be a positive integer.

- The currency values in your output do not have to have cent values of exactly two digits i.e. values such as $0.5, $0.0, $2.5 are acceptable

A note on the style of your source code:

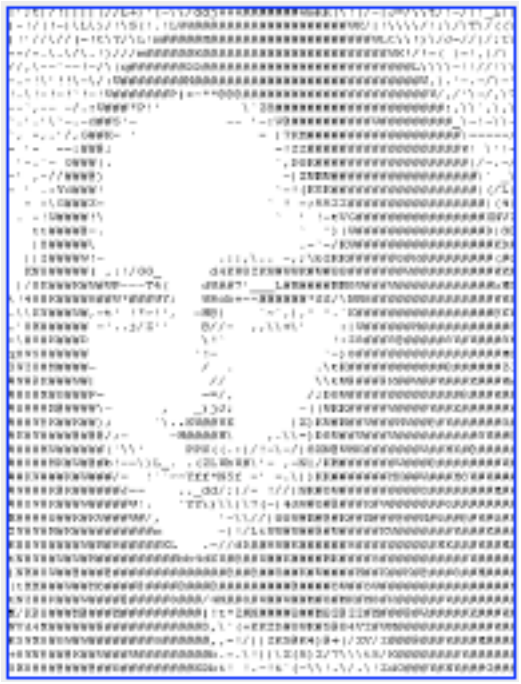
- there should be no numeric literal values *anywhere* in your source code, unless they are being assigned to symbolic constants. Five symbolic constants have been defined for you. If you need to use a numeric literal value, assign it to a symbolic constant first and then use the name of the constant in your source code.

NOTES:

- Your program must be called `Q1Program.java`
- You will also need the `Keyboard.java` file for reading keyboard input
- To run your program, you should download the application class called `Q1Application.java`
- Your `Q1Program.java` file **must** include a comment at the very top that states your name and describes the purpose of the program
- Be very careful with the formatting – your output must be formatted exactly as shown in the examples above (spaces, punctuation and capital letters should be in exactly the same places)
- Make sure you only use features of the Java language that have been covered in lectures – remember that your work will be reviewed by your peers
- Read the marking schedule (attached at the end of this handout) carefully to make sure you have done everything required to get full marks for this question

Question Two (8 marks)

For this question, you must write a program which produces an *ASCII art* drawing. Such a drawing is one which consists only of text characters that can be entered at the keyboard. Several, quite large, examples are given below:



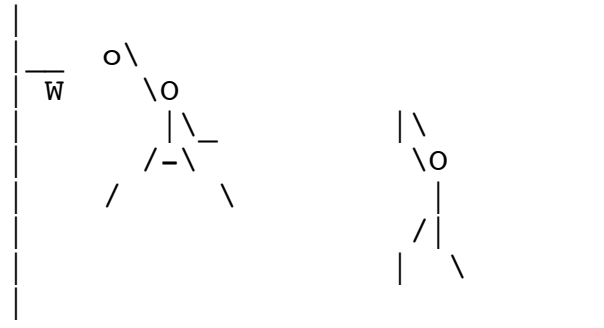
Although the examples above are fairly complicated, your ASCII art drawing can be much simpler. You can draw absolutely anything you like, as long as your drawing consists of **at least** 10 lines of output. Your drawing **must** include at least one character that requires an escape code of some sort (e.g. `\n`, `\\`, `\"`)

There must also be **two** additional lines of output, printed ***after*** the 10 lines or more for your picture. The first line of output must print a title for your picture (the title may help the marker see what your picture is supposed to be!).

If you have copied the picture or parts of the picture from the internet or any other source, the second line of output must be the word "Source:" followed by where you obtained the picture from.

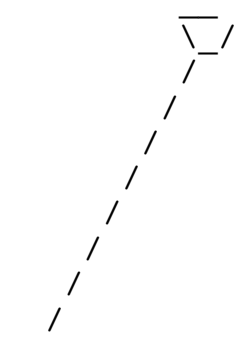
If you have created the picture completely by yourself, the second line of output should be the word "Original".
A couple of examples are shown below. In the first example, a source for the picture has been acknowledged, and in the second example the picture is original:

```
c:\Asst1> java Q2Application
```



```
Two people playing basketball
Source: www.scopie.com
```

```
c:\Asst1> java Q2Application
```



```
A nail
Original
```

There are no extra marks allocated for artistic ability. All of the above drawings would receive full marks.

NOTES:

- If you created the picture yourself, remember to display the word "Original". Otherwise, remember to acknowledge the source.
- To run your program, you should download the application class called `Q2Application.java`
- Your `Q2Program.java` file **must** include a comment at the very top that states your name and describes the purpose of the program
- Read the marking schedule (attached at the end of this handout) carefully to make sure you have done everything required to get full marks for this question

SUMMARY OF SUBMISSION INSTRUCTIONS:

- You should submit the following SIX files for this assignment through the web dropbox. Refer to the Chapter "How To Submit Assignments" in your course book for details.

    - `Q1Application.java` – your application class for running the program for Question 1
    - `Q1Program.java` – your program which calculates the cost of a toll call.
    - `Q2Application.java` – your application class for running the program for Question 2
    - `Q2Program.java` – your program which produces the ASCII art drawing
    - `Keyboard.java` – the provided class for reading keyboard input
    - `A1.txt` – a text file containing your feedback on the assignment (see below).

- The Java source files (`Q1Program.java` and `Q2Program.java`) will be reviewed by your peers. The text file containing your feedback (`A1.txt`) is confidential and will not be seen by your peers.

WHAT YOU SHOULD INCLUDE IN THE `A1.txt` FILE:

- You **must** include a text file named `A1.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?`

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. Therefore, if you are going to resubmit, make sure you resubmit *every* file again.
- Make sure that the files you submit are the final versions of the ones you want 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 penalised 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.

Assignment One Marking Schedule

Please read the marking schedule below carefully. The assignment you submit will be reviewed by your peers and marked according to this schedule. For each question, there is a *Style* and a *Correctness* section.

This assignment is out of a total of 30 marks. It is worth 2% towards your final grade.

Question One (18 marks)

Style

Examine the Q1Program.java file and review the following style categories:

Indentation

0 marks	**Inconsistent** – there is at least one place where the lines of code are not correctly indented according to the standard code conventions
1 mark	**Perfect** – all of the code is indented correctly according to the standard code conventions

Comments

(Note: the author information will be automatically filtered out of the comment at the top of the program by the peer review system)

0 marks	**None** – there is no comment at the top of the file clearly describing the purpose of the program (ie. a description of what it is that the program does)
1 mark	**Poor** – the comment appearing at the top of the file contains spelling mistakes or uses language which is unprofessional
2 marks	**Perfect** – a comment appears at the top of the file which clearly describes the purpose of the program

Descriptive variable names

0 marks	**Poor** – names chosen for at least two of the variables do not describe the information which is stored in the variables. If it is not possible to figure out what the variable is used to store simply by looking at the name, then the name is not adequate
1 mark	**Good** – the names chosen for most variables describe the information which is stored in the variables, however, there is one name which does not describe the information stored in the variable
2 marks	**Perfect** – the names chosen for all variables are excellent and describe what the variables are used for

Variable identifiers

0 marks	**Violate conventions** – there is at least one variable declared which has an identifier that violates the variable naming conventions (i.e. it violates one of the rules that all variable identifiers begin with a lower case letter, all subsequent words which make up the identifier should start with a capital letter, and all other letters should be lower case)
1 mark	**Perfect** – all variable identifiers adhere to variable naming conventions (i.e. all variable identifiers begin with a lower case letter, all subsequent words which make up the identifier should start with a capital letter, and all other letters should be lower case)

Use of symbolic constants

0 marks	**Poor** – there is at least one place in the source code where a literal numeric value has been used where a symbolic constant could have been used in its place
1 mark	**Perfect** – symbolic constants have been used instead of literal values throughout the source code for all numeric data

Correctness

Compile the Q1Application.java and Q1Program.java files and run the program using the Q1Application class. Make sure the Keyboard.java file has been compiled in the same directory (so that input can be read from the keyboard).

Test Case 1 (check formatting and values):

Enter the value **XXX**. The output should be *identical* to that below:

```
c:\Asst1Marking> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: XXX

Number of minutes: XXX
  Twenty minute blocks: XXX ---- $XXX
  Five minute blocks:   XXX ---- $XXX
  Single minutes:       XXX ---- $XXX
---------------------------------------
Cost of call $XXX
```

NOTE: *In this assignment handout, the values that your program will be tested on has not been given to you, but instead is shown as "XXX".*

When you do the actual peer review, you will be told exactly what values to test the program with and exactly what the output should look like.

Format of spacing

0 marks	**No output** – the code does not compile, or the code compiles but no output is produced
1 mark	**Incorrect spacing of characters** – the characters are not spaced exactly as they should be. There should be two spaces at the beginning of the three lines showing the twenty minute blocks, five minute blocks and the single minutes. The number of blocks and their corresponding costs should be lined up.
2 marks	**Perfect** – the format of the spacing of the characters is perfect

Format of characters

0 marks	**No output** – the code does not compile, or the code compiles but no output is produced
1 mark	**Spelling mistake** – there is a spelling mistake in the output (every word in the output should be spelled *exactly* as in the example above)
2 marks	**Missing punctuation** – the spelling of all words in the output is correct, but either the quotation marks around the words "CALL PRICER", or the colons, or the dash marks between the number of blocks and their corresponding cost are missing.
3 marks	**Perfect –** the format of the characters in the output is perfect

Values

0 marks	**No Output** – – the code does not compile, or the code compiles but no output is produced (no values are displayed)
1 marks	**Partly correct** – the number of blocks and their corresponding cost and the total cost of the call are displayed but one or two of the values are incorrect.
2 marks	**Perfect** – all the values are correct

Test Case 2 (check values only):

Enter the value **XXX**. The output should be *identical* to that below:

```
c:\Asst1Marking> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: XXX

Number of minutes: XXX
  Twenty minute blocks: XXX ---- $XXX
  Five minute blocks:   XXX ---- $XXX
  Single minutes:       XXX ---- $XXX
---------------------------------------
Cost of call $XXX
```

Values

0 marks	**No Output** – – the code does not compile, or the code compiles but no output is produced (no values are displayed)
1 marks	**Partly correct** – the number of blocks and their corresponding cost and the total cost of the call are displayed but one or two of the values are incorrect.
2 marks	**Perfect** – all the values are correct

Test Case 3 (check values only):

Enter the value **XXX**. The output should be *identical* to that below:

```
c:\Asst1Marking> java Q1Application
Welcome to "CALL PRICER"
Please enter the number of minutes: XXX

Number of minutes: XXX
  Twenty minute blocks: XXX ---- $XXX
  Five minute blocks:   XXX ---- $XXX
  Single minutes:       XXX ---- $XXX
---------------------------------------
Cost of call $XXX
```

Values

0 marks	**No Output** – – the code does not compile, or the code compiles but no output is produced (no values are displayed)
1 marks	**Partly correct** – the number of blocks and their corresponding cost and the total cost of the call are displayed but one or two of the values are incorrect.
2 marks	**Perfect** – all the values are correct

Question Two (8 marks)

Style

Examine the Q2Program.java file and review the following style categories:

Indentation

0 marks	**Inconsistent** – there is at least one place where the lines of code are not correctly indented according to the standard code conventions
2 marks	**Perfect** – all of the code is indented correctly according to the standard code conventions

Comments

(Note: the author information will be automatically filtered out of the comment at the top of the program by the peer review system)

0 marks	**None** – there is no comment at the top of the file clearly describing the purpose of the program (ie. a description of what it is that the program does)
1 mark	**Poor** – the comment appearing at the top of the file contains spelling mistakes or uses language which is unprofessional
2 marks	**Perfect** – a comment appears at the top of the file which clearly describes the purpose of the program

Correctness

Compile the Q2Application.java and Q2Program.java files and run the program using the Q2Application class. You will have to examine the source code for the Q2Program.java file to check that an escape code has been used correctly in one of the print statements.

ASCII Art

(Note: the artistic quality of the picture is not worth any marks)

0 marks	**Less than 10 lines of output AND no escape code** – either the code does not compile, or the program prints a picture which consists of less than 10 lines of output and no escape code has been used to draw the picture
1 mark	**Either less than 10 lines of output OR no escape code** – either the picture consists of at least 10 lines of output but no escape code has been used, or the picture does include an escape code but consists of less than 10 lines of output
2 marks	**Perfect** – a picture with at least 10 lines of output is printed, and at least one escape code occurs within a String used to draw the picture

Title and source

0 marks	**Neither a title nor a source appears following the picture** – there is no title or source displayed below the picture
1 mark	**Either the title or the source is missing** – a title is printed below the picture but there is no source, or the source is displayed but there is no title.
2 marks	**Perfect** – both the title and the source are printed below the picture. The source is either the word "Original" or an acknowledged source.

Peer Review (4 marks)

You will be awarded up to 4 marks based on the quality of your peer reviewing. Make sure you review each assignment correctly by selecting the most appropriate option for each category and pay attention to the additional comments you provide your peers as feedback. These comments should be constructive – your aim is to help your peers learn.