Assignment 1

Due Date

This assignment is to be completed in pairs. The assignment is due at 3PM Wednesday August 10th 2022 and should be completed with a partner. You and your partner should work together on all of the design and programming. It should be done using the *pair-programming* methodology and not by division of labour. Use the "Assignment 1 Pairs" groups on MyLO to find and register your pair:

- if you already have a partner — start at the bottom of the list of pairs and when you find an empty group, both you and your partner should enrol in that group.
- if you do not have a partner — start at the top of the list of pairs and when you find a group with 0 or 1 member(s), add yourself to that group. If you are the second person to join, you now have a partner and so email your partner to get started.

Background

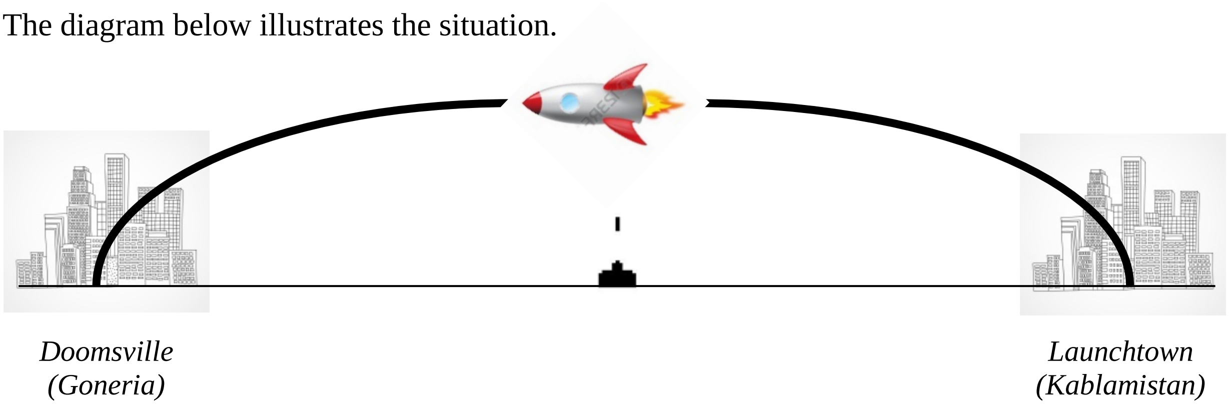
The belligerent country of Kablamistan has launched an inter-continental ballistic missile towards a country 8000km away which threatens to destroy the peaceful Goners who live there.

Fortunately the United Nations has anticipated this possibility and, out of concern for the citizens of Goneria, has placed a high-powered laser at the midpoint between the site of the missile's launch (Launchtown, Kablamistan) and the target (Doomsville, Goneria). The laser is capable of destroying the missile in mid-air — even at its highest point of 20km above the Earth.

Unfortunately the laser's guidance system got fried when the soldier monitoring the system blew its fuse when trying to cook a pop-tart with the laser and no replacement component could be found. Given the military's well known policy of awarding contracts to the lowest bidder, you've been given the job of developing software to upload into the 80286 computer that will be used to drive the laser! For free!

Also unfortunately, the missile is shielded against lasers and will need to be hit multiple times before being destroyed.

The diagram below illustrates the situation.



The technical specifications are as follows:

- The missile
  - is fired towards Doomsville at 12:30PM, i.e. at lunchtime in Launchtown!
  - flies an elliptical path (with a major radius of 4000km and a minor radius of 20km) which is 7998km long. And
  - travels each of the 180 x 1° segments in which it can be targeted by the laser in 7.405s.
- The laser
  - is 4000km between Launchtown and Doomsville directly below the path of the missile;
  - requires 15.737s to charge before it may be fired;
  - moves in 1° increments — where 0° is along the ground to Launchtown, 90° is straight up, and 180° is along the ground to Doomsville;
  - requires 3.61s to move 1° (and starts perpendicular to the ground, i.e. at 90°);
  - should only be fired at degrees in the range 1–179°;
  - may move while being charged; and
  - fires instantly for an infinite distance.

Given an angle `t` in the range of 0–180° for the direction of the laser, the position of the missile on its elliptical path is the coordinate `(x,y)` where `x=4000*Math.cos(r)` and `y=20*Math.sin(r)`. `r` is the value `t` converted to radians and the conversion can be done with a built-in method: `r=Math.toRadians(t);`

If the missile will be at this coordinate at the time the laser is aiming there then this is a *firing solution*. Firing solutions consist of:

- the time the laser should be fired;
- the degree at which the laser should be aimed; and
- the resulting coordinate at which impact with the missile would occur.

Program specification

You and your partner must write a Java program to calculate and display all possible firing solutions. These should be found from the greatest distance from Doomsville to the least distance from Doomsville with the previous firing solution as the starting point for the next.

Your solution should comprise the following files:

- `TimeInterface.java` and `Time.java` — the *Time* ADT as used in lectures and tutorials, extended with a method to add seconds to an existing time. The heading for this additional function should be:
  ```
  void addSeconds(double s);
  ```
  Only the whole number portion of the `double` value should be added to the time object;
- `FiringSolutionInterface.java` and `FiringSolution.java` — the specification and implementation (respectively) of a firing solution. A firing solution should be implemented as a class with instance variables which consist of:
  - the time the laser should be fired (as a `Time` value);
  - the degree at which the laser should be aimed (as an `int`); and

    - the resulting coordinate at which impact with the missile would occur (as two `double` fields, `x` and `y`).

  and the operations should include a getter and setter for each field; and

- `AssigOne222.java` — the file which contains the `main()` method and other functions which implement the required task. (There is no need to sub-divide `main()` into additional methods.)

A starting point is available on MyLO for you to download. You should complete all the required files. You may need `nextLine()` and `nextInt()` from `java.util.Scanner`, `print()` and `println()` from `java.lang.System`, `compareTo()` and `charAt()` from `java.lang.String`, `valueOf()` from `java.lang.Integer`, and `format()` from `java.text.DecimalFormat`. You will also need to use `if` and `if-else` statements, and `for` and `while` loops.

An example of the program's output is shown below:

```
At time 12:33:42 PM, fire at 30 degrees to impact missile at (3464.102, 10.000)
At time 12:34:04 PM, fire at 33 degrees to impact missile at (3354.682, 10.893)
At time 12:34:26 PM, fire at 36 degrees to impact missile at (3236.068, 11.756)
At time 12:34:48 PM, fire at 39 degrees to impact missile at (3108.584, 12.586)
At time 12:35:11 PM, fire at 42 degrees to impact missile at (2972.579, 13.383)
At time 12:35:33 PM, fire at 45 degrees to impact missile at (2828.427, 14.142)
At time 12:35:55 PM, fire at 48 degrees to impact missile at (2676.522, 14.863)
At time 12:36:17 PM, fire at 51 degrees to impact missile at (2517.282, 15.543)
At time 12:36:39 PM, fire at 54 degrees to impact missile at (2351.141, 16.180)
At time 12:37:02 PM, fire at 57 degrees to impact missile at (2178.556, 16.773)
At time 12:37:24 PM, fire at 60 degrees to impact missile at (2000.000, 17.321)
At time 12:37:46 PM, fire at 63 degrees to impact missile at (1815.962, 17.820)
At time 12:38:08 PM, fire at 66 degrees to impact missile at (1626.947, 18.271)
At time 12:38:30 PM, fire at 69 degrees to impact missile at (1433.472, 18.672)
At time 12:38:53 PM, fire at 72 degrees to impact missile at (1236.068, 19.021)
At time 12:39:15 PM, fire at 75 degrees to impact missile at (1035.276, 19.319)
At time 12:39:37 PM, fire at 78 degrees to impact missile at (831.647, 19.563)
At time 12:39:59 PM, fire at 81 degrees to impact missile at (625.738, 19.754)
At time 12:40:22 PM, fire at 84 degrees to impact missile at (418.114, 19.890)
At time 12:40:44 PM, fire at 87 degrees to impact missile at (209.344, 19.973)
At time 12:41:06 PM, fire at 90 degrees to impact missile at (0.000, 20.000)
At time 12:41:28 PM, fire at 93 degrees to impact missile at (-209.344, 19.973)
At time 12:41:50 PM, fire at 96 degrees to impact missile at (-418.114, 19.890)
At time 12:42:13 PM, fire at 99 degrees to impact missile at (-625.738, 19.754)
At time 12:42:35 PM, fire at 102 degrees to impact missile at (-831.647, 19.563)
At time 12:42:57 PM, fire at 105 degrees to impact missile at (-1035.276, 19.319)
At time 12:43:19 PM, fire at 108 degrees to impact missile at (-1236.068, 19.021)
At time 12:43:41 PM, fire at 111 degrees to impact missile at (-1433.472, 18.672)
At time 12:44:04 PM, fire at 114 degrees to impact missile at (-1626.947, 18.271)
At time 12:44:26 PM, fire at 117 degrees to impact missile at (-1815.962, 17.820)
At time 12:44:48 PM, fire at 120 degrees to impact missile at (-2000.000, 17.321)
At time 12:45:10 PM, fire at 123 degrees to impact missile at (-2178.556, 16.773)
At time 12:45:33 PM, fire at 126 degrees to impact missile at (-2351.141, 16.180)
At time 12:45:55 PM, fire at 129 degrees to impact missile at (-2517.282, 15.543)
At time 12:46:17 PM, fire at 132 degrees to impact missile at (-2676.522, 14.863)
At time 12:46:39 PM, fire at 135 degrees to impact missile at (-2828.427, 14.142)
At time 12:47:01 PM, fire at 138 degrees to impact missile at (-2972.579, 13.383)
At time 12:47:24 PM, fire at 141 degrees to impact missile at (-3108.584, 12.586)
At time 12:47:46 PM, fire at 144 degrees to impact missile at (-3236.068, 11.756)
At time 12:48:08 PM, fire at 147 degrees to impact missile at (-3354.682, 10.893)
At time 12:48:30 PM, fire at 150 degrees to impact missile at (-3464.102, 10.000)
At time 12:48:52 PM, fire at 153 degrees to impact missile at (-3564.026, 9.080)
At time 12:49:15 PM, fire at 156 degrees to impact missile at (-3654.182, 8.135)
At time 12:49:37 PM, fire at 159 degrees to impact missile at (-3734.322, 7.167)
At time 12:49:59 PM, fire at 162 degrees to impact missile at (-3804.226, 6.180)
At time 12:50:21 PM, fire at 165 degrees to impact missile at (-3863.703, 5.176)
At time 12:50:44 PM, fire at 168 degrees to impact missile at (-3912.590, 4.158)
At time 12:51:06 PM, fire at 171 degrees to impact missile at (-3950.753, 3.129)
At time 12:51:28 PM, fire at 174 degrees to impact missile at (-3978.088, 2.091)
At time 12:51:50 PM, fire at 177 degrees to impact missile at (-3994.518, 1.047)
```

Algorithm

`AssigOne222.java` could implement the following algorithm:

- setting up the launch time
- iterating through each position (degree) of the missile from Launchtown to Doomsville and for each
    - repeating the following steps until a solution is found or it is determined there are no solutions
        - calculating how the laser would need to move to get there
        - calculating how the missile would need to move to get there
        - determining how many seconds the movement of each item would take and, if the laser can get there no later than the missile
            - determine the (x,y) coordinate of impact and the time
            - create a firing solution of these values
            - display the time and firing solution (you can make use of the defined `DecimalFormat` objects to format the output of `Time` and `FiringSolution` objects)
        - otherwise
            - moving the missile to the next location (degree)

Program Style

Your program should follow the following coding conventions:

- `final` variable identifiers should be used as much as possible, should be written all in upper case and should be declared before all other variables
- variable identifiers should start with a lower case letter
- every `if` and `if-else` statement should have a block of code (i.e. collections of lines surrounded by `{` and `}`) for both the `if` part and the `else` part (if used)
- every `do`, `for`, and `while` loop should have a block of code (i.e. `{}`s)
- the keyword `continue` should not be used
- the keyword `break` should only be used as part of a `switch` statement
- opening and closing braces of a block should be aligned
- all code within a block should be aligned and indented 1 tab stop (or 4 spaces) from the braces marking this block
- instance variables should be used sparingly with parameter lists used to pass information in and out of functions
- local variables (excluding loop counters) should only be declared at the beginning of methods (either as parameters or otherwise)
- commenting:
    - There should be a block of header comment which includes at least
        - file name

        - student names
        - student identity numbers
        - a statement of the purpose of the program
        - date
        - the percentage of the work completed by the authors — 50:50 is expected and assumed but reasons should be given if it is more/less than this
    - Each variable declaration should be on a single line and should be commented
    - There should be comments that identify groups of statements that do various parts of the task
    - Comments should describe the strategy of the code and should not simply translate the Java into English

Marking scheme

Task/Topic	**Maximum mark**
Program submitted in form required	
Program submitted correctly	1
Program compiles without error and runs to completion	1
Program operates as specified	
`Time.java` correctly completed	2
`FiringSolution.java` correctly completed	9
`AssigOne222.java` correctly completed	11
Program Style	
Does not unnecessarily repeat tests or have other redundant/confusing code	3
Uses `final` variables where appropriate	2
Uses pre-existing methods (`format`/`nextInt`/`nextLine`/`compareTo`/`valueOf`/`charAt`/etc)	2
Uses correctly the Java naming conventions	2
Local variables declared at start of methods (other than loop counters)	2
Alignment of code and use of white space makes code readable	2
Always uses blocks in branch and loop constructs	2
Meaningful identifiers	2
Header comments for the program (author, purpose, date, filename etc)	2
Each variable declaration is commented	3
Comments within the code indicate the purpose of sections of code (but DO NOT just duplicate what the code says)	2

What and how to submit

What to submit

- You should submit the three source files (`Time.java`, `FiringSolution.java`, and `AssigOne222.java`) file. That is all that is required. *Do not submit a RAR file — it will not be marked.*

How to submit

- Log in to MyLO and navigate to the `Assignments` tool under the `Assessments` menu in the top tool bar.
- Select `Assignment 1` from the list of available drop-boxes.

- Click on `Add a File` and follow the instructions to attach your source code files and then click `Add`. Then click `Submit`.

If you want to re-submit, please just do so.

Please note: only one submission is required for the pair. You cannot submit as an individual — only as a registered member of a pair. If you can't see the Submission drop-box, it's because you're not registered in a pair. Fix that problem first.

Plagiarism and Cheating:

Practical assignments are used by the School of ICT for students to both reinforce and demonstrate their understanding of material which has been presented in class. They have a role both for assessment and for learning. It is a requirement that work you hand in for assessment is your own.

Working with others

One effective way to grasp principles and concepts is to discuss the issues with your peers and/or friends. You are encouraged to do this. We also encourage you to discuss aspects of practical assignments with others. However, once you have clarified the principles *of the question*, you must express them in writing or electronically entirely by yourself in your pair. In other words you and your partner must develop the algorithm to solve the problem and write the program which implements this algorithm yourselves. You can discuss the question with others, but not the solution. Assistance with the solution should be provided by staff during tutorials or consulting times.

Cheating

- Cheating occurs if you claim work as your own when it is substantially the work of someone else.
- Cheating is an offence under the Ordinance of Student Academic Integrity within the University. Furthermore, the ICT profession has ethical standards in which cheating has no place.
- Cheating involves two or more parties.
  - If you allow written work, program print-outs, or electronic versions of your code to be viewed, borrowed, or copied by another student then you are *an equal partner* in the act of cheating.
  - You should be careful to ensure that your work is not left in a situation where it may be used/stolen by others.
- Where there is a reasonable cause to believe that a case of cheating has occurred, this will be brought to the attention of the unit lecturer. If the lecturer considers that there is evidence of cheating, then no marks will be given to any of the students involved and the case will be referred to the Head of School for consideration of further action.

Julian Dermoudy, July 25th 2022.

Revisions:

- Laser charge time increased to 15.737s