CPSC 233: Assignment 4

Documentation	**Actual score**	**Max score**
Contact information provided in banner documentation e.g., name, ID, course info, tutorial number etc.		1
List of program features completed		2
List of program limitations		1
Inline documentation: lists features implemented in each method		1
Subtotal: Documentation marks		**5**

Style	**Actual score**	**Max score**
Good naming conventions for identifiers (variables, constants, methods)		2
Layout and appearance of source code		2
Layout and appearance of output		1
Deduction: variable class attributes not made private		-1 per attribute
Deduction: static variables employed (except for debugging flags)		-1 per variable
Subtotal: style marks		**5**

Functionality marks	**Actual score**	**Max score**
Displays required contact information as a 'banner'		1
Correctly implements `matches()` in class `Person`		3
Correctly implements `matches()` in class `Location`		3
Correctly implements `matches()` in class `Time`		3
Adding accessors and modifiers in class `Person`		1
Adding accessors and modifiers in class `Location`		1
Adding accessors and modifiers in class `Time`		1
Correctly implements the constructor(s) in class `Detective`		1
Correctly implements `toString()` in class `Detective`		1
Determine the murder information (the murderer, the location, and the time)		9
Determine the murderer with the optimal algorithm[1]		2
Determine the murder location with the optimal algorithm[2]		2
Determine the murder time with the optimal algorithm[3]		2
Subtotal functionality marks earned		30
Deduction: Static methods implemented – aside from `main()` (half functionality marks)[4]		
Deduction: Only a single class is used to implement the solution (half functionality marks)[4]		
Deduction: change the signature of the methods in the skeleton code of the `Person`, `Location`, `Person`, and `Detective` classes (half functionality marks)[4]		
Deduction: input doesn't come from the file with the name specified as a command line argument[4]		
Subtotal: Functionality marks		30

1 For the sample input files "oracle.txt" and "suspects.txt", the optimal algorithm can derive the murder by examining one suspect and querying the Oracle three times.
2 For the sample input files "oracle.txt" and "suspects.txt", the optimal algorithm can derive the murder location by examining two suspects and querying the Oracle three times.
3 For the sample input files "oracle.txt" and "suspects.txt", the optimal algorithm can derive the murder time by examining one suspect and querying the Oracle one time.
4. These penalties are cumulative if all four of these requirements are missed then the functionality marks is divided by 16 (rounded down)
Note: The optimal algorithm will be presented in the second tutorial (Mar 20 or 21)

TOTAL SCORE											40
GRADE POINT											4.3
Raw score	38 – 40	36 – 37.5	34 – 35.5	32 – 33.5	30 – 31.5	28 – 29.5	26 – 27.5	24 – 25.5	22 – 23.5	20 – 21.5	<= 19.5	<= 17.5
GPA	4.3	4.0	3.7	3.3	3	2.7	2.3	2.0	1.7	1.3	1.0	0.0