General questions:

[Coding style and language features]
Q: What do you mean by coding style and language features? I don't want to lose marks so I am worried I might not meet these criteria.
A: Don't worry too much about this. Just follow the minimal number of conventions I mentioned in the lectures, such as give meaningful names to auxiliary variables and functions; use indentation properly; and if a function is too complex, provide comments to explain what you want to do there.

[Computation accuracy]
Q: In the older version of the coursework document, there was a text about returning the values up to 2 decimal accuracy. How can I do that?
A: We have removed that requirement from the newest version now (please update). So don't worry about it.

Q: In the test data pdf, some results are rounded up. I received very similar results but there's still a difference (probably rounding errors). Will this cause a problem (e.g., losing marks)?
A: No, we won't be that strict.

[Cross reference]
Q: Can I import functions from Part A to Part C? I have implemented something in part A, and would like to reuse it in part C, but I don't want to implement again. Can I do this import?
A: While technically you can do, we strongly recommend you not to do so. The reason is there won't be much overlapping (only the calculation of the moving average). So copy pasting the code from part A to part C would make your code less messy (and easier for us to mark).

[Exception handling]
Q: Do I need to implement the exception handlers for Parts C and D as well?
A: No, you don't need to.

[Marking]
Q: What happens if my code fail a test during marking? Will I lose the whole mark there?
A: Even if your code fail, you might not lose a mark, because we will check the code, and if in principle it works well, we will give you the mark.

[Packages allowed to use]
Q: Can I use packages such as pandas or datetime that were not covered in the lectures?
A: Unfortunately, you cannot use pandas and other modules which were not covered in the lectures. The reason for this is that in python there are many useful extra packages that would make the coding very easy. We would like to test your knowledge of programming, not how well you know python. Also, we would like to make the comparison fair between all of you, as many of the students don't know python at all.

[Python version]
Q: Which version should we use, Python 3 or Python 2?
A: Python 3

[Submission]
Q: Do I need to submit the provided skeleton files?
A: Yes, we will import those files to evaluate your code's results (e.g., from parta import <function_name>)

Q: Can I submit auxiliary files such as a new module for my classes?
A: Yes, you can submit additional files.

[Time conversion]
Q: The newest coursework document suggests a new way to do the time conversion between epoch timestamp and human readable date, using calendar. But I have implemented the conversion in the old way, and it works well. Is it a problem?
A: No, if it works well, then it won't be a problem. The conversion described in the document is just a recommendation.

Part A:

[Moving average]
Q: To calculate the moving average of the average daily price, I sum up the total volumefrom, then I divide by the total volumeto. Is this correct?
A: No, this is not correct. You will need to calculate the daily average for each day first (volumeto/volumefrom). Then you take the moving average of these daily average values.

Part B:

[File not found case]
Q: Where should I catch the file not found case?
A: You should implement it in the __main__ part of PartB.py, as Python will throw an error there before it gets to any of the

[Messages to print]
Q: I have caught the error and printed the required message. Shall I let the system print out something else as well?
A: No, it should only print out the required message (see the answer of the next question for the solution).

[Stop or continue after catching the error]
Q: What happens after I caught the error and print the required message to sdtout? What else should I do?
A: You can either return None or call sys.exit(). Both will let the code terminate swiftly. We are only evaluating how you can catch the errors. Therefore you can quit the program after you have caught it.

Part C:

[Using out-of-range data]
Q: In order to calculate the moving averages of the dates given within the interval, can I use the data from dates preceding the interval?

A: Yes you will have to. For example, the moving average of the first day requires the knowledge of daily averages of the days before that day, so they are out of range.

[Signal for the first day of the range]
Q: In order to generate a signal for the first day, I would need to calculate the short and long moving averages for the day before as well. Do I have to do that?
A: No we don’t expect you to do that. The main reason is that it’s already too much work you have to do, and doing that extra calculation doesn’t make your code better or worse Note that in practice you should cover this case as well, but for now just ignore it.

Part D:

[Classify trend]
Q: Can I use the slope of the daily high and daily low data to do the classification?
A: Yes, this is exactly what you we expect you to do.

Q: If I have to calculate the slope of high and low values, which interval I should use?
A: Use the dates between start_date and end_date of the Investment instance.

[First 5 functions]
Q: Am I required to use the first 5 functions in the last 2 (predict_next_avg and classify_trend)?
A: No, you don’t have to.

[Missing arguments]
Q: When I implement the first 5 functions of class Investment, I realise that the class may already store the data and the start and end dates. Can I assume that I don’t have to pass these arguments to the function?
A: Yes, you will need to be able to handle both cases: when all the arguments are passed to these functions, and when some are missing. In the latter case, the function should use the corresponding variables of the class instance instead (Hint: use optional arguments).