Overview

This activity aims to teach students about the need for consistency and atomic blocks through the use of transactions. This is an introductory exercise. Consistency and atomic blocks are the idea that threads should only be able to access data at the right times and in the correct order in order to complete their task effectively.

Suitable For

Higher/Advanced Higher Computer Science

Key Concepts

The use of transactions, multi-threading, parallelism, basic understanding of arrays, consistency and atomic sections.

Learning Outcomes

- Understand how bank transactions work
- Understand why threads can't access the same data at the same time
- Understand the importance of serialisability (but not really mention it)
- Understand why it is important to have threads reading and writing data in a specific order

Success Criteria

- I will be able to put a set of instructions from two different transactions in order.
- I will understand why operations in a transaction need to be atomic, and understand the meaning of being atomic.
- I will be able to describe a series of transactions in order that makes them atomic

Time Required

1 period - 1 hour

Preparation

1. Print out the handout A and handout B on separate pieces of different coloured paper to show the separation of the transactions.

Prior Learning Assumed

None - this is a new topic, perhaps they should already know about threads and being thread safe

Outline of Activity

1. Outline to the class that in order for banks to keep track of their bank accounts correctly they have to make sure that actions are done in the right order otherwise mistakes are made.
2. Outline that more than one transaction (a series of actions) can happen at one time, for example someone might be withdrawing £10 out of their account at an ATM at the same time that their friend is adding £20 into the same account online. If things are done in the wrong order then mistakes can be made and the bank account could be shown to hold a value of money that isn't correct. These are called read-modify-write

operations, where you read an initial value, then change it in some way, then write back the new result.

3. Explain that a concept called threads are used to do the read-modify-write operations. (Needed???)
4. Using the cut out labels, outline that if the two transactions are interleaved, they can hold different values at the end of the transactions running. Explain that this is wrong and not atomic.
5. By putting the transactions in order, this is atomic and it means that there is only ever going to be one correct value for the bank balance at the end of both of them. If you don’t read or write the transactions correctly then the two threads are going to cause problems within the bank account.
6. Finalise by explaining that this is the way atomic operations are set up, so that you don’t have overlapping values for example mistakes being made in the bank balances.