COMP 4958: Lab 4

Submit a zip file named `lab4.zip` containing the file `arithmetic.ex` (for part 2) and a folder named `card` (for part 1). Note: Do not submit the `_build` directory. Maximum score: 15

1. Implement a supervised card worker (in a module named `Card.Worker`). The card worker is a registered server and its interface is basically the same as the card server specified in lab 2 (i.e., no pid parameter to `new`, `shuffle`, `count` and `deal`), except that `start` is replaced by `start_link`.

   Implement also a "store server" in a module named `Card.Store` that provides functions `get` and `put` to read and write the state of a card worker to a file. Name the file `cards.db`. Use the `--sup` flag when creating the `mix` project. Note that when the card worker is restarted by its supervisor, it should retain its previous state. To faciliate testing of restarts, the card worker

   - must print a message whenever it is being started;
   - should "crash" when `deal` is called with an argument that is not an integer. (Simply don't test for an integer argument.)

2. The `Arithmetic.Server` from part 2 of the previous lab creates a pool of workers to handle requests. We want to improve `Arithmetic.Server` so that when any of its workers dies unexpectedly, the server creates a new worker to replace it. Effectively, `Arithmetic.Server` becomes the supervisor of its workers.

   Make changes so that when a worker dies, the server is notified and starts a replacement worker process. To facilitate testing,

   - do not handle the error that occurs when a worker is asked to calculate the square or square root of a non-number, so that we can crash the worker by asking for the square of, for example, `:hello`;
   - print the new PID of the replacement process when it is created.

   Note that the PIDs of all the workers are printed when the server starts up, as specified in the previous lab.

   Put your code in a file named `arithmetic.ex`. Use what we have talked about in class (exit signals, links, etc.) to implememt this. (Do not use monitors or polling using `Process.alive?`.)