**Computing Course Work Ideas:**

So I need 3 ideas for my computing project for a level.

I am confident that I will have the technical skills to create a variety of impressive projects but I just need to figure out what the best project ideas are. I want to do one that:

* Leverages my skills and experience is the fields of programming I have pursued already
* Leverages my strong maths background in order to try to show of to universities my further maths and decision maths abilities

What project have I completed in the past (fully fledged and finished projects):

* School projects that were easy and boring:
  + Dice game
  + Number converter
  + Chatbot
* Interworks SQL stored procedure converter
* Not yet completely finished: pi weather station backend
* Emailer

What am I particularly proficient with in programming:

* Web technologies: JS and HTML
* Flask to make a rest API of webserver
* Web scraping to gather information of the web
* SQL alchemy and marshmallow and json to allow for data base and data storage solutions
* NumPy arrays for matrix calculation (or use my own library)
* Writing libraries for mathematical processes
  + Matrix calculations
  + More efficient factorial algorithms
* Writing efficient algorithms:
  + Writing efficient sorting and searching algorithms
* Writing code for data structures:
  + May as well show of how I can write the code to implement a stack and a queue
* Interacting with APIs and json data: e.g. scryfall
* Object Oriented Programming
* Recursion

Maths skills that I have been able to relate to programming:

* Matrix operations
  + Markov chains
* Linear programming
* Binomial theorem and infinite series
* Graph operations like Kruskal’s and Dijkstra’s

Would be interested in further research into these topics in order to create a project:

* Simple neural networks
* Tree searching for solutions
* Simplex method of linear programming
* (potentially 3d vectors)
* Game theory
* Matching algorithms
  + Look at patent expired old amazon book recommender algorithm
* Markov chains further
  + Weather prediction
  + Natural language processing and prediction
* Fourier transform and fourier series

What sources of help do I have for research:

* My book on this a level programming project and how to approach it
* Articles online on how to do X:
  + <https://www.kdnuggets.com/2019/11/markov-chains-train-text-generation.html>
* Oxford short introduction to:
  + Machine learning
  + Game theory
* Year 2 further maths books:
  + Simplex method
  + Game theory

What ideas have I already had:

* Freezer inventory system that uses stick on qtr. codes and a database lookup to keep track of what is in the freezer, its expiry data and potentially it could create a restock list, could be adapted to work as a example system that a supermarket may use when doing inventory and shelf management
* Recommender algorithm:
  + Could make a spoof match making site where people complete radio button forms about them and their interests and are matched to others
* Shortest path analysis program that works by web scraping google maps and detecting roads and adding them to a graph. Will provide the shortest path to a given destination
* Amazon price tracker, create a program to web scrape amazon pricing information (hard)
  + Allow users to add an item to a wish list and be alerted by email when the price has fallen to within an acceptable price
* Machine learning project to do image recognition.
  + Start by generating a bunch of similar search queries for sample data
  + Web scraper google and Bing images
  + Make a simple swipe left of right to allow the technician (you) to rapidly remove images that are not suitable for training data
  + Could be used to:
    - Constantly film a bird feeder and only keep recording that are of Robbins or squirrels for example
* Weather prediction: I could create 2 models and test them against using historical weath station data to try to predict the current weather station data / weather. I could use polynomial interpolation etc to predict characteristics like humidity and then use this to generate a sunny / rainy / cloudy prediction. I could use a Markov chain model that uses the last X weather events to try to predict the current weather event.
* I could make a text prediction program using Markov chains
* Hypothesis testing program
* A twitter crawler combined with a person recommender algorithm
  + Could sweep profiles for keywords
  + Recommend new profiles to users with similar interests
* A linked in job listings scraper combined with an alert system to email about new jobs that may be suited to an individual
* Multiplayer uno game that is accessed though a website and allows 2 players to play a game in real time with a chat function with the use of web sockets
  + Opportunities to be simple: start with prototype 1 as noughts and crosses and json data store
  + Use sql database to store the leader board and allow it to be downloaded
  + Email people a challenge that allows them to paly another player.

**Other notes:**

* Uses agile methodology
* GUI is important: usability is important
* Miss cannot give written feedback
* Start iterations during the simmer
* 150 plus pages of work
* Top mark comes form planning
* Maked internally
* Do what you are interested in
* For feedback on the user experience use a person who has not
* You can have a physical computing component
* You only submit one proposal
* It doesn’t have to be a gap in the market, this is only an a level project, it can be a problem that has already been solved

**Questions:**

* What scope is necessary / excessive?
  + Can I make a project with a wish list of extra features
  + Yes this is the idea: this will enable you to pan an ambitious project and will enable you to add to your future features evaluation
* Can you breach some laws: e.g. Twitter bot that breaks data protection act, Uno game that breaks copyright?
  + Yes that is fine
* Are all languages acceptable?
  + Watched craig and Dave video and It didn’t include html and js
  + Yes this is fine
* Does user friendly interface mean that the UI must look good

**So my ideas Generally**

* Most of my ideas fall into these categories:
  + Web scraping
  + Machine learning: writing this off
  + Maths related: the maths won’t wow them so for the most part writing this off
  + Recommendation algorithm: I can do this without AI
    - <https://realpython.com/build-recommendation-engine-collaborative-filtering/>
  + (obviously I will not be doing a game)

**What are my interests that I am not sure how to wrap into a project:**

* Machine learning
* Data science
* Encryption and security

**What ideas am I drawn to?:**

* Amazon price tracker
* Linked in job scraper
* Machine learning bird recogniser app
* (Weather data prediction)
* (Multiplayer game like uno or noughts and crosses)

Now I will try to evaluate the merits of each project idea

**Birds recognition website (for mobile):**

Sources (technical feasibility)

<https://docs.graphene-python.org/projects/sqlalchemy/en/latest/tutorial/>

So I now have talked through some ideas and have some sources to read:

* I talked about a price comparison bot that will web scrape eBay and Amazon. This may not be suitable as it is hard to predict the extend of counter web scraping strategies. I may waste lots of time web scraping and yet it is not that impressive
* A weather simulator, it will use a variety of mathematical techniques in order to try to predict the weather including Markov chains and polynomial interpolation. It could use historical weather data from an api to predict current and future weather then grade itself on how well it did. Users would be able to see a dashboard of the weather predictions
* A bird logger. It would be a ai free version of a previous idea. A sql database with a graphql api will allow users to search for a bird and see info about it. They can then log sightings in their garden.
* A chess program:
  + This will use symbolic reasoning and tree searching to create an ai chess bot
  + Users will be able to play the chess bot on a website
  + Users will be able to login to access higher difficulty and to save with the leader board
    - Email verification
    - Password management and hashes in database
  + Could allow players to challenge other players and play player vs player online with web sockets
  + Opportunities for relational data in sql database:
    - Many to one relationships
      * User games vs AI
      * User leader board entries
  + Lots of opportunities for an iterative development cycle e.g. the first prototype GUI could be a grid containing letters that show pieces and then an instruction box and a coordinate grid input box for entering a move

I will now read the details about the project form OCR:

* I may need to ask permission to use TypeScript
* It is rare that they allow projects that don’t include a GUI
* The focus is on documenting the cycles well and so you can still score well with a smaller (less huge in my case) project scope. While scope can be later refined, it is ideal to have a skeleton that is a minimum that you know you can achieve and then extra features, this will stop you overestimating / putting too much on your plate and help you differentiate your prototypes while allowing all of them to meet the initial requirements
* Even though Miss Flyn-coley cannot give me written feedback I can still get verbal help and feedback in computing club sessions she said.
* It explicitly mentions these ideas as good for the appropriate scope:
  + Simulations: my weather idea
  + Complex games: not really applicable as the goal of the chess program is not to create a complex game
  + Automation and scheduling / timetabling
  + Online multiuser websites (chess program)
* Note first prototype could use a restful api to be later upgrades to graphql
* If I am using web sockets exclusively I could use a node JavaScript webserver and a python sockets server to act as the brains
* Persistent data storage using SQL that is advanced will be seen as impressive
* I will use either a functional or an object oriented paradigm (either is fine)
* You are allowed to use libraries (clear reference required)
* You will not be rewarded for implementing common algorithms and data structures that already exist, you are not rewarded for reinventing the wheel. For example use python lists and inbuilt sorting functions. Your project should require you to code some more complex algorithms (for me this will be the tree search symbolic reasoning logic)
* You should document throughout all challenges, what you did to overcome them. What decisions around implementation where made and why.
* JS (html and CSS should not be the focus / to elaborate), python and SQL are accepted languages
* For a web based projects the final iteration will need to be hosted
* The Craig and Dave document contains a mark breakdown for each step so it will be a good thing to familiarise yourself with as you go, you are aiming to get all 10 marks

List is things to email Miss Flyn-Coley about:

* Using TypeScript which isn’t in the OCR list of languages although JavaScript

These sources encourage you to narrow down to 3 and then make a more detailed proposal for each in order to help you narrow down.

My 3 best ideas currently are:

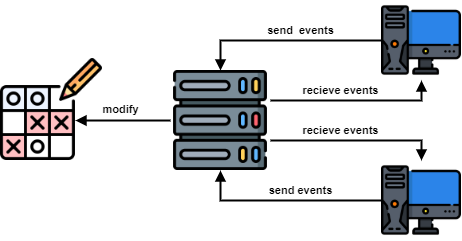
* Weather data simulation / predictor that uses a web interface
* Chess program using symbolic reasoning and a web interface
* (least favourite currently): A bird app that that allow users to access a large database of info about bird and log bird sightings

**Chess program:**

Core skeletal features:

* Allow the user to access the program through a web interface
* Allow the user to play chess against a computer

Advanced feature Wishlist:

* (The following features require a database that could be implemented with graphql and sql-alchemy)
* Allow for the user to login
* Allow for a logged in user to access difficulty levels
* Allow for a logged in user to add there score to a leader board
* Allow for match statistics to be saved to the database and then analysed and recalled to be displayed on the website
* More or Less Rejected: Allow for 2 logged in users to play against each other:
  + Use an api to email a challenge that includes a URL
  + I have looked into this and the best sources seem to be examples of 2 player tic tac toe games: <https://www.programonaut.com/learn-how-to-build-a-multiplayer-tic-tac-toe-2/>. They use web sockets to communicate
  + 
  + I understand the concept but now yet how to implement this, this may be a step to far and an unrealistic feature

Benefits of this project:

* It allows me to demonstrate knowledge of web technologies
* It allow for complex algorithms:
  + Symbolic reasoning by tree searched (depth adjusted for difficulty)
  + All possible moves and following moves at a certain depth will be evaluated and scored. The best move will be picked, then the user will more and the next best move will be re-evaluated
* Actually I found an explanation of how to do a 2 chat in sockets <https://codeburst.io/building-your-first-chat-application-using-flask-in-7-minutes-f98de4adfa5d>

**I just talked further with dad about the chess program: it’s the one**

* I could rename is a board game engine so that my minimum prototype can be draughts with a simple GUI and python backend that decides moves
* I could account for all valid moves of pieces by modelling their possible moves as vectors and allowing only reflection and possible scalar multiples of these
* I could easily create saves of games allowing them to load where they left off. This could be stored in the database
* I could create a rewatch system that records all the moves in the game in a json file allowing the game to be recreated and displayed
* I could do wider reading to learn about ways to advance my chess algorithm. For example heuristics to discard some branches form the tree allowing more in depth searches for the same resources
* You could add a chat function during the user on user chess match

I could make prototypes:

* Symbolic reasoning to win at for example tic tac toe
  + I would need an object that represented a game state before the computers go
  + It would have an attribute which would be a generator of possible moves and the possible board states
  + You could then make a recursive algorithm to recursively search the tree of all moves. Then you would score each possible move by the potential for the computer vs person to win and then persue the best branch
* I could make a multiplayer tic tac toe game
* Graphql and sql database test