[THE PROGRAMMERS IDEA BOOK]

200 Software Project Ideas and Tips to Developing Them

Written by Martyr2 – Version 1.05

This book is dedicated to all the staff, alumni, mentors, experts, authors, contributors and visitors of the Dream.In.Code web forum.

- Your Humble Servant Martyr2

Table of Contents

Preface

Back in December of 2008 I started a thread on the web board DreamInCode.Net (where I mentor) that listed 150 programming project ideas to help address a serious need... “I want to practice programming, but what kind of project should I create?” The question has come up on the board several times before with the typical response listing only a handful of ideas. So to stem the tide, I created that thread which has now reached over 250,000 views and ton of positive responses.

After the tremendous support from those on the board, and the spread of the list on the web, I decided to create this book and grow the list. I have added an additional 50 ideas along with a difficulty rating, tips to help the programmer think about the problem and present ideas for additional functionality.

I intend on making this book a somewhat live document that I will grow over time for all those who purchase it. I plan on adding additional ideas, tips, revisions and possibly tables/charts to help explore those ideas already mentioned as well as the new ones yet to come.

I hope that you take a moment to recognize the effort and time put into this and purchase this book if you haven’t already.

Introduction

Welcome to The Programmers Idea Book: 200 Software Project Ideas and Tips to Develop them. This book was created to inspire programmers of all levels to develop their own program when they are bored or looking to beef up their skills. This book was designed with the idea that each problem should be solvable with any of the popular languages like C#, VB.NET, PHP, Python, Ruby, Java and others. While some languages may be more well suited for a problem, most of the project ideas can certainly be ported over to others with minimal effort.

This book divides 200 program ideas into 10 distinct categories based on the most relevant idea that the project may implement. Some of these projects could easily fit into multiple different topics and be implemented through several different designs. Please keep this fact in mind when creating your solution. Just because a project idea is specified under the category “Files” it may very well have a minor “Web” component to it.

Every project has 5 parts. A title, a difficulty rating, a project description, tips to help the programmer when thinking about a solution and ideas for adding additional difficulty / functionality to the project.

Each project idea has a difficulty rating between 1 and 10; with most projects falling between 4 - 8. These ratings are given purely as a guide for the programmer to determine scope and its possible complexity. A project rated as an 8 will often be much more complex, or require more

research, than a project rated as a 2. It is recommended that beginners start with projects between 1 and 4 and tackle higher level projects as they gain experience.

The tips provided for each project may reflect only ONE way to solve the problem, but certainly not the ONLY way. These tips are meant only to get the programmer into thinking about the problem, possible approaches, or help when the programmer is stuck.

Lastly, the added difficulty section was meant for when a programmer who has successfully implemented a project and has a working prototype (aka an official version 1.0 release). These ideas can be implemented in further releases or spur ideas for additional features. These can also be explored to build additional programming knowledge about the topic or project idea.

I hope you enjoy the book and if you make the next multi-million dollar program based off of one of these ideas, be sure to share it with me.

Numbers

Title: Add/Subtract/Multiply/Divide Fractions
Difficulty: 5
Description: This project involves working with fractions. How do you add ⅓ + ⅕? Create a program that first asks the user which operation they want to do: add, subtract, multiply or divide and then asks for 1, 2 or more fractions to work with. The program prints out the result.

Tips: Think about how you might divide up the fraction itself into its numerator and denominator. Many of the operations involve finding a common denominator, getting its reciprocal and possibly reducing a fraction to lowest terms. This would be much easier if you can isolate the numerator from its denominator. You could also do this using a custom fraction class.

Added Difficulty: Try working with mixed, improper, or complex fractions.

Title: Alarm Clock
Difficulty: 3
Description: Create a program which shows the current time using a label. Allow the user to select a given time and specify a message. When the clock reaches that time, have it pop up a message to the user reminding them of their message and/or play a specific sound file.

Tips: You will need to be able to get the current time and display it to the user. You will also need a mechanism for saving an "event" which has a specific time and an associated message. Try creating a custom class to hold this information. On each tick of the clock, compare the time to each of the saved scheduled events. If one matches, show the associated message. Be sure to check for multiple events for a given time and possibly throw away any old events (expired) in the system if they are not needed.

Added Difficulty: Make the clock display graphical using pictures for the numbers. You could also integrate a database to keep track of multiple events much easier. Add an icon to your messages and possibly specify which sound file is to played as the alarm. Remind the user if an event is "overdue" and by how long.

Title: Binary to Decimal and Back Converter
Difficulty: 4
Description: Make a program which has the user enter in either a decimal value and prints out its binary equivalent or enter a binary value and it prints out the decimal equivalent value.

Tips: Start testing with some low well known decimal and binary values to make sure that it is first converting correctly. Use values you can easily calculate like 3, 4 and 5. Try taking a number, dividing it by 2 (using integer division) and recording its remainder. Do remember that N % 2 is always going to come out as either 1 or 0. Take the result of the divide and divide that by 2 again. Keep doing this until you can no longer divide it by 2 (again using integer division).

Added Difficulty: Try converting the value into Octal or a unit in some other base.

Title: Calculator
Difficulty: 6
Description: Create a simple calculator application which can add/subtract/multiply/divide integer or floating point values. Make sure it includes numbers 0-9, the four operators +, -, *, / and an equals button.

Tips: One way to think about this problem is how you might break the equation down using a stack. Place operators on a stack data structure and pop them off as you go to evaluate two operands together. For instance, 2 + 3 might be parsed by saving 2, place + on a stack, saving 3 and when the user presses equals it takes 3, pops off the + operator, takes 2 and knows to add the two operands together. The reason for the stack is for later where you can handle multiple terms with different operators and keeping order of evaluation.

Added Difficulty: Implement other calculator functions like modulo, factorial and raising a number to a given power. Try adding the ability to add parentheses to control order of operations.

Title: Change Return Program
Difficulty: 2
Description: Create a program where it shows the user a list of items to buy and their price. Then ask the user to pick an item and enter in the amount of money they would have inserted into the “vending machine”. Have the program calculate the change and return it to the user in the form of quarters, dimes, nickels and pennies. For example, if the user chooses an item that costs $1.25 and they say they give it $2.07 the program would print out 3 quarters, 1 nickel and 2 pennies as change for the user.

Tips: First find out the difference between the item they bought and how much they are paying. Then look at using integer division to first calculate how many coins of each type to return. Then use the modulus (% operator in most languages) to determine the remainder of the change and repeat the process for each denomination. Be sure to check that the money they have inserted will first cover the price of the item! For example 89 cents / 25 is 3. This means 3 quarters with 14 cents returned.

Added Difficulty: Create a menu system where the user can quickly select the item they wish to buy. You can also take it a step further and create a full graphic user interface (GUI) showing the front of a vending machine where they can press buttons to specify which item they want or the amount they entered.

Title: Credit Card Validator
Difficulty: 6

Description: How do we know that the credit card number someone entered on our store is correct? How do we know it is valid for the type of credit card they specified? Create a program which allows the user to specify a credit card number and the card type (Visa, Mastercard, American Express or Discover) and return a message if the number is valid or an error message if the credit card number is invalid.

Tips: This program is all based around the idea of the Luhn algorithm (Check Wikipedia for "Luhn Algorithm"). This algorithm can be used to validate credit card numbers. First make sure the number they entered is the correct number of digits and of the correct number prefix (Check out Wikipedia for "Bank card number" for more information). Once you determine the correct prefix and the right number of digits, you can then try applying the algorithm.

Added Difficulty: Accept more card types or determine if the card type is active or not. You could also try wrapping this up into a simple function which you can save in your function library for future use in projects.

Title: Decimal to Roman Numerals
Difficulty: 4
Description: Have the user enter in a simple integer value and it will print the Roman numeral representation of that number. For instance if the user enters 11 it will print out XI. If they input 9 it will print out IX.

Tips: Break the number down by finding the largest round number that will go into it and then translate that to a roman numeral (you can have a list of the constants like "M" being 1000). Then subtract that out of the remaining number and do it again. It is advisable that you take a look at the "Change Return Program" first before attempting this problem. Keep in mind that numerals on the left subtract from those on the right. The value 9 is not VIIII it is 1 subtracted from 10 or IX.

Added Difficulty: Expand the range of supported numbers into ten thousands, hundred thousands and millions. These symbols appear with a horizontal bar over the top of the symbols. If you can't represent them in your language, perhaps come up with your own way of showing them (create your own symbol?)

Title: Dijkstra's Algorithm
Difficulty: 6
Description: The whole purpose of this algorithm is to find the shortest path through a graph of nodes with the least cost. If you are not familiar with these types of graphs, imagine a map where roads connect between major cities. One road from Denver may connect to Los Angeles while another road may connect to Chicago. Chicago can then link to New York etc. Each of these roads have different lengths and thus some take longer to drive than others. That is the cost. So which route is the fastest to get from Los Angeles to New York? Is it via Chicago or is it faster to go through Nashville?

Tips: This is a lesson in recursion where you want to test various routes between your starting point and the end point and find which route between two nodes (called an edge) has the least cost. You may need to go down one node that is more costly than another if it means that the edge following that is significantly cheaper than an earlier choice. For instance, if you have two routes which are 11 and 15 respectively, you may need to take the 15 route if it means that the edge afterwards is only 2 (15 + 2 = 17) where if you had gone down the 11 route the only option after that was 10 (11 + 10 = 21 and is thus longer). So keep track of where you have been and test various routes that you can go. Not all nodes may be visited!

Added Difficulty: Display the solution graphically by drawing the node relationships, their route values and then highlighting the shortest path between point A and point B.

Title: Distance Between Two Cities
Difficulty: 7
Description: Ask the user for latitude/longitude coordinates for two given cities and calculate the distance between those two cities in either miles or kilometers.

Tips: The world is not flat! To make things simpler, use formulae that doesn't take into account ellipsoidal effects. So the recommendation below is based on a spherical Earth to make things easier. First stop to help solve this problem is the Haversine formula on Wikipedia. This formula can be used to calculate the distance between two points. Simply plug in the latitude and longitude and the radius of the Earth (around 6,371 km). Be sure to test with points you know the distance between and can easily look up. This will let you know if you are in the right ball park with your results.

Added Difficulty: Implement the Spherical Law of Cosines to get distances much more accurate and possibly down to around 1 meter.

Title: Fibonacci Sequence
Difficulty: 2
Description: Create a program that calculates the Fibonacci sequence up to the Nth term. Ask the user to enter the Nth term and have the program calculate the sequence until it has printed that many terms.

Tips: There are two main ways you will find people solving this problem. There is the recursion method (which is a nice example to learn about recursion but a horrible practical application to solve it) or using a simple loop that keeps track of the two previous numbers to add them together to find the next value in the sequence. This second method is the more efficient method for computing because it doesn't require multiple calls going on the call stack.

Added Difficulty: Try implementing both ways so you can see the difference between the two. Provide a graphic explaining why the recursion method may be more resource intensive than the preferred loop method.

Title: Find Cost of Tile to Cover W x L Floor
Difficulty: 1
Description: Ask the user to enter in a width, length and the cost per 1 unit of flooring. Have the program calculate how much it would cost to cover the area specified with the flooring.

Tips: This is a relatively simple program. Be sure to first find out how much area the floor is and then multiply that by the cost per unit of flooring. Start with some simple numbers that you can quickly calculate in your head. Try a 10 x 10 ft room with each unit of flooring costing $1.00.

Added Difficulty: Calculate how much flooring would be needed for non-rectangular rooms. Also figure out how much labor costs would be given that the average flooring team can only put in 20 square feet of flooring per hour at a cost of $86.00/hr.

Title: Find PI to the Nth Digit
Difficulty: 3
Description: Create a program where you ask the user to enter in the number of digits to calculate PI to. The program then prints PI up to that many digits.

Tips: There are many ways to do this from the simple solution of implementing the series 4/1 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + … to using the BBP formula discovered in 1995 which calculates PI in base 16 which you could then convert. Try the simple approach first to estimate PI and loop until the desired precision is reached.

Added Difficulty: Use the BBP formula which you can find on Wolfram MathWorld.

Title: Household Budget Program
Difficulty: 8
Description: Create a full GUI program that allows the user to enter in and setup a household budget. They can enter in unlimited number of budget categories like Utilities, Travel Expenses, or Child Care. Then let the user enter in X number of sub-items for each of these categories listing their monthly expenses along with any income items. Have the program keep track of their expenses and their cash flow. This program should let the user know if they are overspending or how much they are saving for the month or year.

Tips: This program was rated with a difficulty of 8 due to the amount of time that can be put into this program, the complexity of setting up an appropriate design and framing out the structure. It is suggested that you start with a few simple categories with a few sub-items for each and sketch them out on paper. Then try to figure out the appropriate classes/structures that you can use for this along with the forms needed to create categories, items and entering costs. Think

about how you might go about summing up items for each category (arrays? loops?) and calculating revenue vs liabilities.

Added Difficulty: Generate some reports based off of the entered data. You could also keep track of numerous years or create a snapshot of a certain time period that can be used in comparisons.

Title: LCD/GCD Least/Greatest Common Denominator
Difficulty: 3
Description: Create a program that asks the user to enter two fractions. Have the program find the least common or the greatest common denominator between those two fractions and print it out.

Tips: Make sure to isolate the denominator first and find a multiple that goes into each of the denominators. You can start by finding the least common multiple (the smallest integer value that fits into both denominators). The greatest common denominator is the largest integer value which fits into both denominators.

Added Difficulty: Try working on more than two terms.

Title: Mortgage Calculator
Difficulty: 5
Description: If you take out a loan from a bank of $10,000 with an annual interest rate of %5 percent, how much is your monthly payments over a term of 36 months going to be? What if you pay $50 dollars more a month, how long will it take you then to pay it back then? Create a program which asks the user for a loan amount (principal), the interest rate and the term. Make it generate a payment schedule showing how much the user has to pay per month over the course of the term to pay it all back.

Tips: Keep in mind that this isn't simply taking the loan amount and dividing it by the term. Interest is accruing and that means that after your first month, you actually have to pay back more than $10,000. As the user makes payments, they are also paying the interest that is growing on that debt. Use some simple numbers to test and make sure that on the final month the balance reaches 0, not a negative number. Banks don't pay you! The final month's balance may not be a full payment.

Added Difficulty: Allow the user to specify various term types. Perhaps they can specify years, perhaps they can specify how many months or days. Build in different rates of compounding the interest. A simple addition would also be to show the user at the end how much they really paid over the course of the entire loan.

Title: Next Prime Number
Difficulty: 2

Description: Ask the user to enter a number and have the program find the next prime number after that value. If you don't remember, a prime number is a number which can only be divided by 1 and itself. Thus if the user enters 3, have the program find the next prime number (5) and print it out.

Tips: There are many ways to find primes. One of the simplest ways to find a prime is to simply start from the number the user enters and loop through the numbers one by one testing if it is prime. You can test if it is prime by trying to divide that number by all numbers starting from 2 until the square root of that number. If none divide evenly, it is prime. For example, to test if 11 is prime, we take the square root of it (which is a little over 3) and we test from 2 to 3. If none of them divide evenly (test the remainder of the division) then it is prime. Since 2 nor 3 divide evenly into 11, it is prime. Test 12... square root is again a little over 3 so we move from 2 to 3 and see if any divide evenly. Since 2 divides evenly into 12, it is not prime. Look at the modulus operator to help you do this.

Added Difficulty: See if you can speed up this method by checking the last digit of the number. Obviously if the number ends in 2 it is even so it not going to be prime (unless the number itself is 2).

Title: Pascal's Triangle
Difficulty: 4
Description: Create a program that asks the user for a number greater than 1. Then construct a Pascal's triangle with that many rows. Pascal's triangle is a pyramid of numbers where the number immediately below the two above it is the sum of those two numbers. Refer to Wikipedia for Pascal's Triangle for great examples of how this triangle will look.

Tips: Think in terms of loops here for a moment. Each row can have its values calculated by following a formula like value = previous_column_value * (((row + 1) - column) / column) and then multiplying it against the previous column's value. Here we know that each row starts with 1, so plug this into the equation above and loop through all the columns of a given row. So the value of the second column of the third row is going to be 1 * (((2 + 1) - 1) / 1) which equals 2 and multiply that by the previous value of that row (1) and we know the result is 2.

Added Difficulty: Make sure you handle the spaces to form a very nicely formatted triangle. Have the program generate the triangle as a graphic.

Title: Prime Factorization
Difficulty: 3
Description: "Prime Factorization" is finding which prime numbers multiplied together make the original number. For instance, the value 12 has the prime factors 2 x 2 x 3. All the factors (2 and 3) are prime. Ask the user to enter in a number and find all the prime factors of that number. Print them off.

Tips: The trick here is to start with the lowest prime number and work our way up. This means we start with 2. Does 12 divide evenly by 2? Yes, 2 x 6. So we know one of the factors is 2. Now we take 6 and again try dividing by 2. Yes, works again. Now we have 2 x 2. This leaves us with 3. Does 2 divide into 3? No. So we try the next prime number which is 3. Does 3 divide evenly into 3? Yes. Thus we know that the prime factors are 2 x 2 x 3. To help with this process, think of a loop which can divide the remaining number by primes. It may help to first create a "isPrime" type of function.

Added Difficulty: Build a chart showing various numbers and their factors.

Title: Sieve of Eratosthenes
Difficulty: 4
Description: This is another algorithm for finding prime numbers up to a specific integer. This is based off of the idea of listing the various numbers and, starting at 2 (which we will call "p"), crossing off all numbers in the list which are multiples of 2. So crossing off 4, 6, 8, 10... up to the integer the user enters. Then go to the next non-crossed off number (in our instance "p" is now 3) and cross off all the multiples of 3. Keep doing this until the value p^2 is greater than the integer input by the user. Create a program which asks the user for an integer and prints off all the prime numbers up to that number using this method.

Tips: Create a list. This list can be an array or an arraylist or a vector. It is up to you. Loop through the array looking for which of those are easily divided by the current number ("p") and mark them as "deleted". This can be done by setting the value to null, calling remove() or use some custom implementation. When p^2 has been reached, simply loop through the list once more printing those which have not been deleted. Don't forget to make sure your list is sorted first!

Added Difficulty: If you used an array, try an arraylist or vector. If you used an arraylist or vector, try using an array. You can also try the variation on this algorithm called Euler's Sieve. Check out Wikipedia under "Sieve of Eratosthenes" for more information on both Sieve variations.

Title: Tax Calculator
Difficulty: 1
Description: Create a program which asks the user to enter in a dollar value and the current tax percentage. Then print out the subtotal along with the total plus tax (grand total).

Tips: This is a very simple program where you collect the value from the user, have the user enter in a tax value, add the tax to the total and print out the result. If the value the user enters is $1.00 and the tax is 5% then the total is $1.05.

Added Difficulty: Build a GUI for this program where the user can enter in multiple prices, adds them up and then calculates the tax. Perhaps some of the items won't be taxed based on the item type (like food).

Title: Unit Converter (Temp, Currency, Volume, Mass and More)
Difficulty: 2
Description: Develop a simple program which asks the user to first pick a conversion type (temperature, currency, volume, mass or some other unit type) and then asks the user for the unit of measure to convert from and to convert it to. Lastly, ask the user to enter in a value which the program then converts to the target unit. For instance, the user specifies temperature and then specifies Celsius to Kelvins. They then enter in a value for Celsius and the program converts it to Kelvins and prints the result.

Tips: Start with a simple unit conversion process like Temperature and try numbers which are easy to convert like 0 Celsius which is 32 in Fahrenheit. Be sure to find the formula that can convert the value from its source to the target value. In this case F = C(9/5) + 32. You may want to quickly locate all the formulas needed for the program first and then translate them into programming formulas.

Added Difficulty: Allow the user to specify how to format the output. Perhaps they want to specify it to 2 digits or 3 significant figures. Perhaps you can make a program variation of this dedicated to a specific profession. A converter for air conditioning and heating repair workers?

Text

Title: CD Key Generator
Difficulty: 3
Description: Create a program which generates keys for an application that you may put on a CD. Classic example would be the keys you use for installation of a major software product from Microsoft or Adobe. Have the user specify the length of keys and the types of characters they can use in the key (only letters, letters & numbers, just numbers, any special characters). It will then generate a random key value that can also be verified. In addition, create the mechanism for validating the generated key.

Tips: This is all about generating random strings using a known procedure or algorithm. There are many algorithms out there used to generate hashes or random passwords. This would be no different EXCEPT you need to know how it is created and then know how to decipher/validate it. Start with a small string and a known encryption algorithm that is easy to decipher and build up from there. Something as simple as a Ceasar or Vigenere cipher could be used to encrypt a special word with a salt (like the current time). Then decipher it to see if that works. Once you have a model working, you can then build off of that with a more complex procedure.

Added Difficulty: Try linking this newly generated key to a specific binary version. For instance a key you generated for version 1.0 of your product would then not work for version 2.0.

Title: Check if Palindrome
Difficulty: 2
Description: A palindrome is a string of characters when read from left to right and right to left is exactly the same (ignoring any spaces usually). One of the most well known palindromes is “race car”. No matter which direction you read it, it still reads “race car”. Create a program which checks if a string entered is a palindrome. If it is, print “The string ‘_____’ is a palindrome” where the blank is the string they entered.

Tips: Simple approach, take the string, reverse it and compare. Depending on the language, you might want to take special precautions in first making sure things like spaces are stripped out. Also make sure you are comparing the strings value, not if two strings are the same object! (This goes for you Java users)

Added Difficulty: If you hadn’t done it already, write the reversing function yourself. Don’t rely on a version provided by the language.

Title: Code Skeleton Creator
Difficulty: 7
Description: Make a program which generates the skeleton or “stubs” for a given programming language. Have the user select which language they are using and perhaps specify a list of class, function and method names (and their parameters/return types) to then have the program generate those items automatically. Make sure that the user can’t specify any parameter names, patterns etc that the language doesn’t support. For example, you may be able to specify an asterisk in C++ for a pointer argument, but the user shouldn’t be able to put that into a Java function template.

Tips: Depending on how you collect the information and the language chosen, you could have a templated system in place. Since most classes, functions and methods have a set format signature you could simply take in the info, fill in a function or class template and then write that into the file. As for validating their input, perhaps this would be useful to run against a regular expression?

Added Difficulty: Support at least 4 languages which 2 are interpreted and the other two are compiled.

Title: Count Vowels
Difficulty: 2
Description: Make a program which asks the user to enter in a string and then prints out how many vowels are in that string. For example, the user enters “hello world” and it returns “There are 3 vowels in ‘hello world’”.

Tips: Strings are typically thought of and used as an array of characters. If you can loop through these characters one by one, you can compare them to a list of vowels. If you find one that matches, increment a counter variable. After the loop is done, the counter will then contain your count.

Added Difficulty: Count the consonants as well as spaces. Provide a mini report of all these counts.

Title: Count Words in a String
Difficulty: 2
Description: Write a program which asks the user for a string and then counts how many words are in that string. For example if they write “This is my first program” the program would print out “There are 5 words”.

Tips: Depending on the language you are using, this can be wickedly simple or a bit more difficult. For languages that have a built in split function, you can split the string on the spaces and then simply count the pieces in the resulting array. If the language doesn’t support a split, you will need to loop through the string and when a space (or end of the string) is encountered, you know you are at the end of a word. For each word you can increment a counter.

Added Difficulty: Along with word counter, count the number of sentences and paragraphs.

Title: Font Viewer and Tester (Online or Not)
Difficulty: 7
Description: Write an application which can be used to view all fonts installed on a given system. Alongside each one provide a sample of that font. This allows the user to browse their installed fonts or possibly view fonts from a font website quickly and easily. Make the program able to show the sample text in a given font size and style. For instance, allow the user to see a sample text not just in Arial, but also as Arial Bold 26pt.

Tips: This is geared towards GUI applications and may be extremely difficult if not impossible to show on some console applications. Some languages (like .NET or Java) provide built in font handling objects and methods. So if you need help try taking a look at those first. PHP users can mimic this by specifying CSS styling for their HTML output.

Added Difficulty: Allow the user to specify the sample text (so they can see what their text would look like in a given font) and let the user choose to automatically install that font if it is not already installed.

Title: Fortune Teller (Horoscope)
Difficulty: 5

Description: Create a program which allows the user to specify their birthday, and based on that information along with today's date, tell them their horoscope. This could be generated from a list of predefined horoscopes or pulled from a web site service.

Tips: The birthday will tell you what sign they are and in turn will tell you which pool of predetermined responses to pull from. You don't want to be generating a horoscope for an Aries that they then see again later for a Gemini. You could have these horoscopes in a text file and read at random which you then display to the user. If you are deciding to use a web site, then be sure to look at the type of information they expect for input and make sure that info is collected from the user. Most of the time horoscope sites will return several paragraphs of text, but be ready to also receive images and formatting.

Added Difficulty: If you are using a text file method, allow the user to add some horoscopes of their own. To do this ask them to specify the sign and then the horoscope message. If you are using a web site, allow the user to change the web site source in case one site is shut down they can change it. If the site provides an image, show it.

Title: Guestbook / Journal
Difficulty: 6
Description: Develop an application where the user can keep track of journal entries or create a public guestbook where multiple people can write in it to say "there were here". This application lends itself perfectly for online sites but could certainly be done as a stand-alone application. It is also possible that this guestbook / journal could reside on a network and people can access it from multiple computers. The program should keep track of the date and time of a post, allow the user to browse through various days, edit/delete offensive posts or queue posts before showing publicly.

Tips: This is a classic application in information management. How you store this information will be dependent on the technologies you use or the scalability you desire. It could be a simple text file, it could be a huge database or perhaps somewhere in between. Have the interface take in the post information (perhaps an author's name) and their message. Make sure you validate the post and render its information harmless. One way to do this is by substituting dangerous characters (look up the concept of "code injection" if you are unfamiliar with the idea) or escaping it.

Added Difficulty: Build a back end admin panel to manage posts more effectively. This may require you to also design a login system that can tell the difference between ordinary users and admins.

Title: Morse Code Maker
Description: 4

Description: Make a program which takes in a string from the user and then outputs that as a string of Morse Code. If you are unfamiliar with Morse Code, you can find it on Wikipedia.

Tips: This is all about mapping a character to its Morse code equivalent. A string is seen as a list of characters in many programming languages so start by looping through these characters one by one and looking them up in a table of characters you create. Map each character to a Morse Code character. That way you can quickly identify which code goes with which letter and print it to screen. Displaying Morse Code characters may require you to draw them on screen rather than simply printing. Do a little research into this area before proceeding.

Added Difficulty: Translate the code generated back into the original text. This may require you to keep track of where spaces were also placed. Try taking a file as input, translating it and printing it back to another file.

Title: News Ticker and Game Scores
Difficulty: 6
Description: Create an application which links to major sporting web sites (or web services) and pulls down current news headlines and game scores then scrolls them across the desktop or web site.

Tips: First of all decide which sources you want to pull from, how they are formatted and if you will need to page scrape them. Sometimes you can save yourself some headache if you can find RSS feeds or XML formatted web services which you could quickly parse. This may be a bit more difficult to do with languages like C/C++ since you may have to deal with sockets directly. However, this should be quick and painless in languages like .NET, Java or Ruby/Python. Try making two easily accessible and iterable lists. One will service as the most current list to be displayed on screen and the other will serve as a buffer for upcoming news/scores. This second list is the one you can manipulate and get ready to put into the active list.

Added Difficulty: Load the program with several sources and make sure, when scores/news are displayed, they are displayed in proper groups. Some examples of groups would be: World news, US News, NHL scores and NBA scores etc.

Title: Pig Latin - Simple
Difficulty: 4
Description: Ask the user to enter in a string which will then be printed back to screen in Pig Latin. If the user enters “hello world” it will be printed back to them as “ello-hay orld-way”.

Tips: First break the string into words. This can usually be accomplished by some sort of split function. Validate the first character of each word and simply apply the rules of Pig Latin. If it is a consonant, strip it off the word and append it to the end with “ay”. If it is a vowel, simply attach “way” or “ay” (depending on the variant you wish to use). Be sure to also check for consonant clusters like “qu” and move them both if need to. Validate the word first and then decide what

approach to take to that word. Join the words back together after all transformations have been made. I would ignore analyzing compound words for syllables.

Added Difficulty: Analyze words to see if they are compound words and then apply the appropriate rule to those types of words.

Title: Post-it-Notes Program
Difficulty: 6
Description: Make a program which a person can enter in a reminder message and have it display on their desktop or on a website. Allow the user to enter as many messages as they want and it will easily organize all the messages in a post-it-note type of fashion for easy reminders.

Tips: You need to be able to keep a list of each message and figure out a way to display them to the user quickly and easily. Keep in mind that the reason post-it-notes are nice is that they can be stuck to things to provide notices, warnings, reminders, dates and times etc. Make sure that your program provides a way to also preserve this easy viewing and DOES NOT INTRUDE on the user's ability to do any other tasks. Perhaps this means putting an icon in the system tray or creating an extra ticker bar at the bottom of the screen.

Added Difficulty: Try making each message its own yellow sticky note looking background graphic.

Title: Quote Tracker (Market Symbols etc)
Difficulty: 8
Description: Write a program which fetches the current market data for a given set of companies (using their market symbols) and scrolls that market data across the user's screen. Various pieces of data can be displayed including: current share price, daytime high/low, volume, last trade date and perhaps a graph to show it. Ask the user for many different market symbols and a particular stock exchange to pull the data from. Then have the tracker periodically update the user on the status of the stock.

Tips: First identify a source for the information and this will determine the type of application you can build. One of the easiest sources of information is from Yahoo Finance which offers a quick URL for downloading stock data as a comma separated file (CSV). You can check it out at the following address http://www.gummy-stuff.org/Yahoo-data.htm. Download the data, parse the fields and then use them to build out the tracker.

Added Difficulty: Have an option to track a particular company by name instead of its symbol. You can also have the user enter in multiple exchanges that the company may be on (such as TSX as well as Nasdaq). Then keep track of the company stock over all those exchanges.

Title: Random Gift Suggestions
Difficulty: 4
Description: Make a program which asks the user to input people's names and the various gift ideas they would eventually like to get for them. Ask the user to also input various important dates related to the people they input (like an anniversary, birthday, Christmas etc). When a given date rolls around, have the program remind the user of the date and show them the gift list they have created for that person.

Tips: This is easily handled by first creating a class representing a person. The person class will contain a list of gift suggestions for that person. It could also manage the list of important dates for that person. Then as time progresses, the program can check all Person classes and see if any have the given date listed for them. If so, read back the list of gifts to the user. Obviously the program is also going to keep track of a list of Person class instances and a timer to check them periodically.

Added Difficulty: Make it so that after the list of gift suggestions are shown to the user, the user can click one and it will open up a web browser and places they can possibly buy that gift.

Title: RSS Feed Creator
Difficulty: 5
Description: Develop an application where a user can enter in some data (be it a blog post, some news article, headline etc) and it formats and publishes it as a RSS feed which then others can subscribe to and read.

Tips: Before doing anything first look up how a RSS feed is formatted and which fields are required vs optional. Then look at the element's relationships and how the various pieces fit together. Design a mechanism for retrieving each piece of information needed from a given source and then turn to your language's XML functionality to construct the RSS document tree.

Added Difficulty: Make the program able to export in either RSS 1.0, 2.0 and/or Atom. Also make it connect to a third party site and publish the feed to it which then others can use in their RSS readers.

Title: Recreate Grep (Pattern Matching)
Difficulty: 7
Description: Grep has been a long standing command in the world of Linux/Unix. Develop your own grep tool. Look at Grep on Wikipedia for basic usage, flags and format.

Tips: This is going to be all about regular expressions. Start simple and work to more complex formats. One of the easiest tests is to be able to specify a word or phrase, and possibly a file list to, then print all lines which contain that word or phrase to that file.

Added Difficulty: See how many flags and formats of the original grep you can implement. Can you also specify piping? (Controlling where the output goes... to screen, file, etc)

Title: Regex Query Tool
Difficulty: 8
Description: Create a tool that allows the user to specify a string and then a regular expression to match against it. It will then return all matches it finds as output. This can be done within a form where then each match can be highlighted in the string itself. This will help developers who want to know, on the fly, if a string can be matched by a given regular expression.

Tips: As the user types in the regular expression, check if the string is already entered. If it is, go through the string and highlight any matches that are found it in with the regular expression. Do this for each keystroke so that as the user builds the regular expression they can see the results instantly. Great for testing regular expressions. Try to find, in your language of choice's regular expressions implementation, a match array or collection which could certainly help you not only identify the first match, but multiple matches if they are present. Also look into creating a stand along function used for highlighting the text. Obviously it will have a start position and end position which you will feed it based on a given match.

Added Difficulty: Give the user many options including case insensitivity and different regular expression engines they can use (.NET, JavaScript, Perl).

Title: Reverse a String
Difficulty: 1
Description: Create a program that asks the user for a string of input and simply returns it in reverse order. For instance they enter "Hello" and it returns "olleH".

Tips: Keep in mind that a string is often treated as an array of individual characters. In C/C++ based languages this list ends with a terminating character '\0'. One of the fastest ways to reverse a string is simply to swap each character as you work your way inward. So for instance, you would swap "H" with "o" and then "e" with "l" etc. To do this, use a loop that continues until the start meets or surpasses the end. Make sure not to swap the terminator itself!

Added Difficulty: None

Title: Text Editor
Difficulty: 5
Description: Make a simple notepad style editor that the user can use to create, open and edit simple text based files. Provide functionality found in your typical editor including saving, copy/paste and printing capability.

Tips: Start with something simple and work your way up. Create the application to first be able to open a window that will allow the user to type into it and save. This window may contain a

multiline textbox or textarea (if you are doing a web app), a toolbar and a status bar. Think about how you would read the file into this new window and write it back out. Once you have the basics of reading in and writing out done, you will be able to then do your saving and opening dialogs and from there to copy/paste and printing. Think about how your editor may react if someone tries to open a file which is not text based.

Added Difficulty: Add in formatting options like bolding, italics and underline. You can also add in a tabbed interface that will allow the user to open multiple text documents at the same time and edit them individually.

Title: Text to HTML Generator
Difficulty: 6
Description: Come up with a simple program that will allow a user to enter in various bits of text data. Have it then generate the HTML equivalent or create a HTML page based on the data. The idea here is to give the user the option of writing simple text data and it would generate the code afterward. For instance, if the user enters in a title and a few paragraphs, the program would generate the necessary <h1>, , <quote>, bulleting tags needed to format that text on the web.

Tips: Most tags in HTML are used for wrapping text (aka markup the text). Think of a paragraph and how to show one on HTML. You would wrap this text in tags. The trick is to parse what they have specified in the program, identify the pieces like a paragraph or the title and then wrap it in the appropriate tag. To help you do this, you may want to check out regular expressions. To identify a paragraph, you could look for text that has double spaces before and after it. Maybe even simpler yet is looking for double spacing.

Added Difficulty: Support more complex formatting like alignment, indenting, styles and inserting certain multimedia objects like video or audio.

Title: Vigenere / Vernam / Caesar Ciphers
Difficulty: 5
Description: Want to get into the basics of encryption? Well a great place to start is with these three ciphers. Create a program which implements these three encryption methods. Ask the user for a string, or file to read, and have the text run through each of these ciphers. Let the use choose whether to output the data to the screen or back to another file.

Tips: First stop, Wikipedia to learn how these ciphers work. Start with the Caesar cipher which is essentially a shifting of the alphabet. Take the letter “a” and shift it down 4 and you have “e” and a “b” becomes the letter “f”. Once you have this program working, then you can move onto the Vigenere cipher which uses shifts to come up with a more elaborate system of encryption.

Added Difficulty: For fun, apply multiple shifts to the same text and keep track of the shift steps. Send the steps to a friend along with the encrypted text and have them see if they can decipher it using the reverse shifting.

Title: Write Out Number
Difficulty: 4
Description: Make a program which asks the user for an integer and then writes out the number in plain English. If they enter in 101, it would print “one hundred one” to the screen.

Tips: We can’t make a direct translation of this due to the placement of each number. So what we must first do is identify each number and its place. Is it in the tens? Hundreds? Thousands? If we take our example of 101, we can identify that the first 1 is in hundreds place by using the modulus operator and integer division. Once we isolate it as hundreds, we can then translate that as “one hundred” and the leftover 1 is identified to be a single value and so it can be just “one”.

Added Difficulty: Support floating point values. The decimal place can be referred to as “and” just like you would write on a personal check. 101.34 would then be “one hundred one and thirty four”.

Networking

Title: Chat Application (IRC or MSN Style)
Difficulty: 9
Description: Create an IRC chat client (like mIRC) or a MSN style client so that the user can communicate with their friends on an established network. Make sure it can connect to a server, maintain a session on that server, maintain one or more network connections and send and receive messages.

Tips: This can easily be a project which sounds simple enough until you get into the specifics of working with the protocols and handling multiple connections. First read through the protocols needed to chat on your chosen network. One protocol may label a private message with the word “privmsg” while another may use a completely different keyword. You can then use this to setup a private message window (if it doesn’t already exist) and put the message there. Make sure you write out your design on paper and that your design is going to work for multiple different IRC servers on different networks.

Added Difficulty: There are tons of additions you can do including: sending files, support for coloring text, voice/video chat, built in scripting functionality or support more networks.

Title: Country from IP Lookup
Difficulty: 6

Description: Develop an application which can look up the country an IP is from. For instance, if a given IP address is traced to a .ca domain it would tell the user the IP is from a Canadian ISP.

Tips: All users on the Internet have an IP address. They are given this IP address from their service providers who were given a range of IP addresses. This means that using the IP address of a user we can find which ISP was given that address range. Then from that we can find the country. You may need to convert this address into an actual IP number. Then with the number we can ask a database like ip2country for the country.

Added Difficulty: Once the country has been identified, identify the ISP it belongs to and where they are located. You can also go one step further and provide this graphically on a map of the given country much like Google Analytics does for stats tracking.

Title: Email Server
Difficulty: 8
Description: Write a program that turns a standard computer into an email server. Using the POP3/SMTP protocols, write the software to accept email, store it into user mailboxes, handle connections from clients and manage email messages.

Tips: Email is simply messages saved for specific users. Those users fetch emails using the POP3 protocol (for our purpose) and email is sent using SMTP. Read the appropriate RFC documents and focus on one user fetching one or more email messages. Then expand it to handle more accounts. Pay special attention to the idea of mime types and multi-part messages which are often sent these days and can contain images and other files. These files are also encoded in base64 so look into that as well if you wish to send images.

Added Difficulty: Allow users to relay messages, store attachments and specify their own ports for both checking or sending email.

Title: Fetch Current Weather
Difficulty: 4
Description: Make a program that prompts the user for a city or zip/postal code and returns the current weather information for the area. Have it specify the temperature in both Celsius and Fahrenheit, daily high/low, wind speed and relative humidity.

Tips: First locate a source of weather data that you can pull from. Find out what data that service needs so you know how your program can query the source and get its information. Analyze the format of its response and then create the program to generate the needed data to submit to the service and how to display results back to the user. Look at national weather center APIs. Google currently has an unsupported weather API as well. Just look up “Google Weather API” for articles on the topic. It is an easy way to test your application.

Added Difficulty: Fetch current weather conditions (sunny, rainy, snow, fog) and show an appropriate graphic to represent those conditions. Add one animation somewhere in the project.

Title: FTP Program
Difficulty: 9
Description: Develop a full FTP program which will allow the user to connect to a host, see the files on that host, download/upload files and make FTP specific commands to change file permissions, delete files, overwrite files and transfer in ASCII as well as binary formats.

Tips: This is a rather complex project when it comes to supporting all the various features. Start with the simple things like connecting to a remote server, logging in and requesting a file list. It is easy to bite off more than you can chew so make sure you create a small demo. Once you have achieved this, then try issuing some more of the complex commands like deleting and overwriting. Lastly, work on the mechanism for establishing a connection to download a chosen file and upload a file from the user's computer. FTP servers may actually pass you an IP address and port it wants your client to change to in order to carry out file downloads. So expect responses that may not be the download itself, but where to go to initiate the download.

Added Difficulty: Handle multiple deletes, moving files in between directories, drag and drop and multiple connection handling through tabs.

Title: Geolocation App
Difficulty: 7
Description: Create a program which takes advantage of the user's location. Using the geolocation tools available in some browsers (such as FireFox or Chrome) identify the location of the user and plot it on a map. Display possible destinations that the user may be interested in in relation to their current location. Assume that perhaps this application would be run on a mobile device and provides latitude and longitude coordinates.

Tips: Many of the major browsers supply geolocation information through an API or by enabling the feature on the browser. FireFox does this through the "navigation" object and more information can be obtained at the following address: https://developer.mozilla.org/en/Using_geolocation. Use this information to then plot where they are on a map. I would suggest using a Google Maps API implementation and from there you can get coordinates of other destinations in the area and plot them too.

Added Difficulty: Tell the user the distance to each of the nearby destinations. You could also possibly show the quickest traffic route to get there.

Title: Get Atomic Time from Internet NTP Clock
Difficulty: 5

Description: Make a quick program which contacts one of the many atomic clock servers on the Internet and retrieves the current time. Display this time to the user and then give the user the option to set their system clock to be in sync with it.

Tips: First we want to find a list of publicly available atomic Network Time Protocol (NTP) clocks online. Try to telnet to one to see if you can contact it and if it will give you the time. Be sure to follow the NTP protocol to communicate with these servers. Once you have a connection established and can see that it is working, try to get your program to connect to it and make the same requests. One place that you can start your research is at the NTP.org website.

Added Difficulty: Display the time graphically on a clock face and have it tick along, making sure to stay in sync every so often (perhaps every 5 minutes?)

Title: Mail Checker (POP3 / IMAP)
Difficulty: 5
Description: Create a program that connects to a mail server and fetches the emails for that user, displaying them in a nice and neat format. Allow the user to list the email headers, read an email, delete one and organize them on the client with folders such as “Inbox” or “Saved”.

Tips: Look up the connection settings needed to communicate with your ISP’s email servers. These are usually published on the ISP’s website somewhere along with the ports needed. Then develop the application to establish a connection to that server and attempt to login using your username and password. Once you are at this point, you are ready to start building out the rest of the application to manage the messages and send commands to the server. Be careful, a server may require you to implement Secure Socket Layers (SSL). Gmail and Hotmail typically require this.

Added Difficulty: Implement drag and drop and if you have developed the email server project listed in this document, use this application to connect to it.

Title: Network Mapper
Difficulty: 8
Description: Develop an application which can crawl the network and discover all the computers on it. Have the program draw a diagram of how the computers are related to one another. For large networks, impose a hop count limit on how far it will map the network.

Tips: This program is very much like Windows Explorer where it can see network computers in your LAN. Recreate this functionality using multicasting which will cross the network and have computers respond back with unicast responses to say “hello I am here”. This should be able to move across routers in the network as well. The diagram could be a type of “fish eye” style of diagram.

Added Difficulty: Enhance the diagram by allowing the user to click one of the computers and it change the diagram to show the network from that node's point of view.

Title: P2P File Sharing App
Difficulty: 7
Description: Ever want to create the next Limewire, Frostwire, Bearshare or DC++? Here is your chance! Create an application that allows a user to login to a file sharing network of peer-to-peer computers and browse/search for files to download or share. Allow the user to chat with other peers, setup the number of downloads accepted, designate a folder to share from and accept network messages.

Tips: Connect to a peer you know is already on the target network and have it send you a list of nodes it already knows about. With this information you can connect to other nodes, query them and send messages to users who are connected to them. This application is all about connection handling so think about how you would handle multiple connections using threading and delegates. The delegates will allow you to communicate between threads and make sure that they are not on the main GUI thread. Otherwise you will have a program that seems to lock up all the time. From time to time, check in with your connected nodes and get updated node lists. You wouldn't want your list of nodes to ever get "stale".

Added Difficulty: Can you create a program that spans multiple different network types? Have your program connect to a Gnutella network and one other network type.

Title: Packet Sniffer
Difficulty: 8
Description: Make a program which tracks packets that come through the user's network. This program will allow the user to see packet information including source and destination as well as payload size and record it for later analyzing.

Tips: First you must capture a device on the network and then read the traffic going to and form the device. This device can be an Ethernet device or another type. If you are developing this application in C/C++ look into libraries such a pcap.h, socket.h and in.h.

Added Difficulty: Build a statistical report that shows how many packets are coming and going, where they are coming from and going to and calculate the speed by which they are moving through the system.

Title: Port Scanner
Difficulty: 5
Description: Develop a program which asks the user for a port number range and then sequentially goes through each port and checks to see if it is open for use. Print on screen which ports are available for communication.

Tips: First of all make sure the number the user enters is in the appropriate range (no negative numbers). Once you have the number range specified, attempt to connect to each of the ports using a socket connection. If the port allows you to connect, then it is open. If not, then it is either in use by an application/service or has been blocked by something like a firewall.

Added Difficulty: See if you can identify which programs/services might be using various ports.

Title: Remote Login
Difficulty: 7
Description: Create a program which will allow a user to remotely login to a target machine. This machine could be a client machine, server, web host etc and be able to do any number of tasks. Perhaps the user wants to just view files, maybe they want to edit or delete files, perhaps they want to be able to have full control of that machine.

Tips: You may simply not be able to reach the target machine if it is behind a firewall and blocks any ports that reach it. This program will require that you have access to both the target machine and the development machine while testing. Make sure you open the target machine enough to first get your client accessing it and after you are sure things are working as expected, you can clamp it down. The types of tasks you do will depend on what the target machine will let you do and what kind of permissions it grants your program.

Added Difficulty: Try making a program that will allow you to control the user's PC and move their cursor, see their screen etc (of course if they give you permission). Can you cause the target machine to reboot? How about run a script?

Title: Site Checker with Time Scheduling
Difficulty: 6
Description: Write a program which will allow the user's computer to check a remote website and see if it is currently up. Allow the user to create a schedule of how often it checks the site from how many minutes/hours/days it was down. In addition, allow the user to add which sites to check and have the program check multiple sites at the same time.

Tips: This project is all about connecting to web servers, requesting a file and if there is a reply that is of the type "200 OK" or equivalent, then the server is up and replying. Make sure you grab a file that you know will always be there (like an index.html file) and follow any 301 redirects if needed.

Added Difficulty: Keep track of the history and chart it to show a graph of the server's up time over the last 30/60/90 days.

Title: Small Web Server
Difficulty: 6

Description: Create a web server application that a user can run on their personal machine. Allow them to set the listening port and manage multiple requests at the same time. Have the application retrieve files on behalf of a user and serve it to them.

Tips: You should take a look at how current servers respond to requests, the types of headers they send and what they expect of clients to locate and specify the type of file they want. This can be obtained in the HTTP RFC. This RFC can be quite extensive, but focus on the basic request and response codes first and then branch out to server other responses and process other types of requests as needed.

Added Difficulty: Implement a server-side scripting process like PHP or ASP. The user would request a file, your server would identify the file extension and pass the file to the parser which would return the result and that data can be sent to the user. Also try to server other media types like images, audio or video.

Title: SMS Email Component
Difficulty: 6
Description: Develop a SMS service that the user can have clients send an email message to and then, using open SMS servers, the application would send the message on to be blasted out to mobile devices.

Tips: On the Internet you should be able to find a list of open SMS gateways by many of the bigger mobile carriers. They often let users send email messages to them for users on their network and your application can take advantage of this. By collecting the info and picking the right gateway, your application would be the middleman in sending out messages to mobiles. Do a search for “open SMS gateways” and you should see some sites that may list many popular ones.

Added Difficulty: Build up the list to include an extensive collection of gateways to cover the globe. Be sure to implement a throttle into the process to prevent users from spamming up the system.

Title: Social Network Manager
Difficulty: 7
Description: Make a program which allows the user to manage all of their social media contacts in one interface. The program should include at minimum: Twitter, Facebook and Linkedin. A user should be able to see current tweets, Facebook comments/likes or messages sent to them privately.

Tips: The trick here is the APIs available from each service. Twitter and Facebook have extensive API frameworks which can help you pull in current information right into your application and display it. Read up on these APIs if you haven’t already and build a prototype for one of them. Then once you have it running, you will have a better idea of how you can

incorporate other services. Keep in mind that you will most certainly have to apply for an application key or be forced to login to get access to certain information.

Added Difficulty: Add in Foursquare and perhaps a YouTube account where the user can watch a video or two recommended to them by a friend.

Title: Video Conferencing
Difficulty: 9
Description: Create an application that can allow two users to communicate with one another via a video feed through their webcams. You may also want to build in a text box where the two people can also type in case their video is having technical difficulties.

Tips: Video is all about speed. We want to send out packets quickly and efficiently while not caring if a packet gets dropped once in awhile. By the time a packet is dropped and the receiver realizes it, the need for the packet has then already passed. For this you may want to experiment with the UDP transmission method rather than TCP. Since UDP is a “fire hose” type of method, it blasts out packets as quickly as it can without needing to do all the checking of TCP. Because of this, UDP is not as reliable but can also help get your video streaming out to the receiver much faster.

Added Difficulty: Try building a picture within a picture so you can see both people during the conversation.

Title: Web Bot
Difficulty: 7
Description: Design a web bot application that will allow the user to crawl website links and index them or do other tasks such as login to a website and carry out various tasks. Some of the tasks may include posting updates to a twitter feed, publishing news, downloading files or scraping pages.

Tips: This type of application lends itself well to threading where the bot can query up a page, fetch out things like links, index them and then start up other threads to follow those links. So dive into the world of threading and delegates for cross threading communication. Keep in mind that sites you connect to will give you a slew of different response headers from 404 to 200 to 500 internal server errors. Be sure to check for each response and act accordingly. Also think about limitations. One limitation may be to not process more than 300 links at a time or follow pages which have already been visited.

Added Difficulty: Index content and create a statistical report of the progress. Graph this progress and make it available to the web bot owner for later review.

Title: Whois Search tool
Difficulty: 5

Description: Have the user enter in either a domain name or IP address and contact a whois service to fetch information. The program should determine who owns the domain, the admin information, DNS servers and contact information.

Tips: Locate a service which will let you specify a domain or IP address, collect that info from the user and pass it along. Then you can receive the response from the service, scrape out whatever data you need to and display it. You may be able to find a web service or REST API itself which will give you this data without needing to scrape a page. Once you find the service, you will have more of an idea of what you need to interact with it and how to format its response.

Added Difficulty: Look up the country where the IP is registered and any other geo-location data you can find about that IP/domain's origin. You could even use a previous project idea to do this.

Title: Zip/Postal Code Lookup
Difficulty: 5
Description: Design a solution where the user can enter in either a zip code (US) or a postal code (Canada or other country) and it returns the city, state/province of that postal code.

Tips: There are many services on the web that allow users to look up a zip or postal code. Have your application interface with one of these services and then pull back the information. You may need to either scrape it off the page (a web page response) or simply parse the response (XML or Web Service).

Added Difficulty: Display the location on a map so the user can visually see where the zip is located within the country.

Classes

Title: Airline / Hotel Reservation System
Difficulty: 6
Description: Create a program which allows a user to take reservations for airline or hotel accommodations. The user can enter a person's name, phone, their length of stay (in case of the hotel example) and assign them to a designated seat or room of their choosing. The system should prevent two different people from booking the same seat/room. In case of the hotel, you can allow double occupancy.

Tips: Think of this in terms of classes and arrays. For the airline example, a plane can be thought of as an array of seats. This array can be an array of Person classes. You would store the person's name and other attributes within the Person class and then store that in the array at the designated seat. If an array slot is empty, no one has booked the seat. In the case of the hotel example, again the rooms can be thought of as an array containing other arrays (2D

array). The array of rooms in rooms[0] may be your $69 dollar a night rooms while rooms in the row rooms[5] might be executive rooms. These room arrays can be filled with room objects which have attributes like “non smoking” or “air conditioning”. The rooms can also contain Person classes. So in other words, 2 Person instances can be in a Room instance which is in the Rooms array on a given room row (like Rooms[0]).

Added Difficulty: If you are doing the airline example, make sure that the person booking can select their own seat or change airplanes. Hotel example, draw out which rooms are available/taken and which are currently being “cleaned”. Schedule cleaning to take 1 half hour in the schedule.

Title: Bank Account Manager
Difficulty: 4
Description: Design a program which acts as an ATM machine. The user can specify a PIN and it shows them a menu of their account types (checking and savings). Allow the user to deposit or withdraw money from a selected account type. Be sure to check that they can’t withdraw more money than they have or if they deposit more than $10k dollars, it lets the user know it will have to contact the bank manager to clear the deposit and won’t let them then withdraw any of that money.

Tips: This is a classic example of class inheritance. Create an Account class which will serve as the basis of both types of accounts. A checking and savings account are both accounts right? They will inherit from the base Account class. Keep basic functionality in the base class like deposit() and withdraw() and then override them in the specific account types. A savings account may also have things like interest applied or a penalty for withdrawing any money. Take this into account when you override those methods. The variable “balance” would also be a protected member of the base class.

Added Difficulty: Allow the checking account to go into negative and apply an overdraft fee of $10. Don’t let the user go into overdraft protection more than $100. You could also prevent the user from making multiple withdraws if they are already overdrawn.

Title: Chart Making Class / API
Difficulty: 8
Description: Create a class or design an API which allows the user to generate specific types of charts like a pie chart, histogram or a bar chart. Ask the user to choose a type of graph they want to make and then collect the necessary information such as axis values, percentages or X/Y coordinates based on the chart they have chosen.

Tips: What you ask of the user all depends on the type of chart they select. One of the easiest charts is the bar chart. For this chart all you need is to know are the axis values and then a value for each bar. Always keep in mind the width and height of your chart when it comes to drawing the values. If you have 500 pixels in width to use for the chart and they enter in 5 bar

values, obviously you are going to have to take the width of the “x” axis, subtract a little spacing between each bar and then divide the rest by 5 (the number of bars). This will give you a rough estimate on the width of each bar. The height will also need to be considered based on the values in the chart.

Added Difficulty: Add full labeling for each chart and try to add one type of animation to it. For example, hovering over a slice in the pie chart will cause it to be highlighted.

Title: Class to Handle Large Numbers
Difficulty: 6
Description: Make a class that can handle large integer values. The user should be able to enter a really large number (like the number of stars in our galaxy or in the known universe) and the class should then be able to store that number as well as allow two instances of this class to subtract, add, multiply or divide values from one another. The result of one of these operations is another instance of the large numbers class.

Tips: One approach is to think of a large integer as a series of numbers much line you would consider a string a series of characters. You can then work on each of these numbers individually as you carry out the various operations. Try working on the subtraction or addition methods first as these should be the easiest to implement.

Added Difficulty: Support some of the other operators such as modulus or see if you can apply various types of formatting to the number via a “ToString” method.

Title: Company Dashboard
Difficulty: 6
Description: Write an application which will allow users of a company to create their very own desktop/online company dashboard. Have the program show things like chosen RSS feeds, blog updates, industry news, website stats etc. Allow the user to customize this dashboard by inserting new “modules” into it, much like iGoogle.

Tips: Think generic here. A widget could be essentially anything but perhaps you would have a widget class defined as a base and then different types of widgets that inherit from it. For instance, perhaps you have a custom widget used for pulling in a RSS feed. Maybe another one would specialize in interfacing with a web page to pull out company news. Another one could be a simple HTML block widget used for displaying static HTML to many users. Create the base class first and what it means to be a widget in your dashboard system. Then you will have a solid class to inherit from and extend for various data sources.

Added Difficulty: Allow the user to interface with public APIs such as Google Maps to show mapping information. Perhaps it can link up with other module information.

Title: Customer Relationship Manager (CRM)

Difficulty: 6
Description: Make a program which will allow the user(s) to create custom records for clients their company may have. The system should allow you to create a new client, insert company information, contact info, titles, general notes about the company or client and what they may have bought in the past etc.

Tips: This is a typical managing program which is best tackled by first identifying what needs to be represented and making a class out of it. Here the client would typically be the class and you would have various properties that each client has such as name, address, phone. Once you have identified the main class that is managed, you would create a second class that manages multiple instances of the first one. Perhaps a "Manager" class would then manage things like a list of clients, sorting the clients, relating two clients together (perhaps a client is a person but that person may also work for another client which is a company). Before coding it all up, think also about how you would go about storing all this in a database. This will help you also figure out the relationships when you do the actual code.

Added Difficulty: Allow the user to add additional yet to be known fields. Perhaps the user is a health care company and they want to be able to track clients, but also know when they had their last vaccination. They could add this as an additional field that all users then have to add when they create a new client.

Title: Employee Time Card program (track overtime, pay, tax deductions)
Difficulty: 5
Description: How much time has Joe worked today? Design a little application which will allow the user to keep track of Joe's work hours and calculate his pay, overtime and any tax deductions that need to be made off of his paycheck. The user will have to enter the user's name, the hours they worked, their hourly pay and choose between zero or many different types of tax deductions (child support etc). The user must also be able to calculate overtime based off a standard work week.

Tips: The class here is the employee. Each employee will have a name, Social Security/Social Insurance number, their pay per hour and perhaps their date of employment. If you create a menu, which will allow the user of the program to quickly identify which employee they want, it will just be a matter of entering in the work hours and calculating their pay based off the employee class' payrate property/method. Overtime is easy to calculate once you know how many hours are in the standard work week. Standard pay is hours worked * hourly pay. But this is only up to the standard work week hours. Once over it turns to time and a half (or some other factor) and becomes hours worked * (hourly pay * 1.5). The 1.5 is time and a half, but double time would be 2 and triple would obviously be 3.

Added Difficulty: Build in vacation tracking into the employee managing program and reject vacation requests if they have not earned enough time off.

Title: Family Tree Creator
Difficulty: 6
Description: Who is related to who? Make a program that answers this question. Allow the user to enter in various people and place them within a tree showing who their parents are, grandparents, sisters, brothers, cousins, nieces, nephews etc. Allow the user to also attach documentation (photos, pdfs, scanned documents etc) to various family members to build off of.

Tips: Here the program is related to the people in the tree. So first think about representing a person with a class and then a class to represent the tree which will manage the relationship between person classes. Two people may marry and have zero to 10+ kids. To help grasp this concept, think about tree data structures which can have more than two nodes (in other words a binary tree won't work). How would you navigate this tree? Take a hard look at recursion as it may help you decide where to add a new member of the family within the tree. Also think about how you might relate two person objects to one another and assign parents or children.

Added Difficulty: Build in the ability to show half brothers and sisters from divorced parents.

Title: Flower Shop Ordering To Go
Difficulty: 6
Description: Women just love flowers. Create a program which allows men to order flowers and have them sent to a specific person at a specific address. The program should allow the user to choose from various flower bundles, specify a delivery address along with a person's name and a small message. Keep track of all orders until the user goes back in and marks them as delivered. Provide a mechanism for searching which orders have not been delivered or have not been processed.

Tips: Ordering systems usually follow one specific relationship... you create an order and on that order you have order items. An order may have things like a unique ID to locate the order for the destination, billing information and a list of all the order items. This is the perfect place to model your order class and how it will go about keeping track of "order item" class instances. An order item will have a unique number and information about a particular flower bouquet. However, it could be part of zero to many different orders so keep that in mind.

Added Difficulty: Give the user the ability to build their own flower arrangement. You may need to then create individual flower items that can be bundled together as an item which will appear as an order item on the order. In other words, 6 roses, 3 baby's breath and 1 vase are all order items that build item 23... the rose bouquet.

Title: Image Gallery
Difficulty: 7
Description: The user may be a starving artist and wants to show their artwork to sell some of their masterpieces. Help them out and create a program which the artist can enter in pictures of their artwork, a title, author name, description and price tag. The program will then allow any

user to then flip through the collection of thumbnails. When they click on one of the thumbnails, it will show a detail screen with the title, description, author name and price tag along with any comments the author may have.

Tips: Think of each item as a class which will consist of the artwork file and various properties like the name, price and description. Keep them in a list and perhaps make them sortable based on a name or price tag. The list itself could be an arraylist, vector or some other type of generic list. When they click the picture, you will fetch this object from the list and simply read the properties of the class to fill out the detail screen. Make sure you take into account what would happen when the reach the beginning or end of the list. Will it simply stop or would it roll to the end or beginning again? Think about how you would manipulate an index value to access each artwork class. Hitting next would increment the index by one, previous would subtract from the index by one.

Added Difficulty: Allow the user to place their favorite items in a favorites list. To do this you will need to build an account mechanism (login/logout) to hold all their favorites.

Title: Josephus Problem
Difficulty: 5
Description: Create a program which asks the user for an integer and then a step value (also an integer). Build a list of numbers between 1 and the first number they entered. This will be a list of people who rather commit suicide than give themselves up to the Romans outside (see "Josephus Problem" from Wikipedia). In a round robin fashion, using the step value, remove a number from the list until there is one number standing. Tell the user that remaining number.

Tips: This isn't too difficult of a situation once you know how to collect the numbers from the user and build an array or linked list that counts from 1 to N. Pick a number at random to start with and then count every step value down the line of remaining numbers and remove it. Keep doing this until there is only one value left in the list. This is your sole survivor... and the one who ends up giving up to the Romans.

Added Difficulty: Use a circular double linked list and remove nodes from the list each time they are removed. Let the user choose which direction to start removing items from.

Title: Library Catalog
Difficulty: 5
Description: Build a virtual library where the user can enter books into their collection, give them an ISBN number and allow other users to then check them out. Build a menu to start where the user can add a book, remove a book, check it out, return it or exit the program. The user can locate a book by title or ISBN and check it out for a specific period of time. They can also return a book. If the return is late, they are charged a fee.

Tips: Here the class can be a book and it may have information like the title, abstract, ISBN and info about whether or not it is currently available. If it is not available, it may also hold a return by date. The add book feature simply constructs the objects and puts it into a list to be managed and the delete functionality removes it. This would typically be done by another class that manages books objects. Would the add and delete functions be part of the book or the managing class? When a user checks out a book you will grab the book object, set its status to unavailable and set its return date. When they return it, take the current date, compare it to the return by date and if it is passed, you know to charge the fee. Otherwise, you set the availability back to available and remove the return by date.

Added Difficulty: Allow the catalog to store a book's picture along with the description.

Title: Matrix Class
Difficulty: 7
Description: Create a class that allows the user to store matrices. Then provide various functions that allow the user to add, subtract, multiply and divide matrices using the object. For example, if we have Matrix A and we want to multiply it by Matrix B, you would do something like C = A * B. Matrix C will be the resulting matrix of the multiplication and another instance of your Matrix class.

Tips: How you do this will differ drastically between languages. Languages that support it can override the four basic operators. For languages that don't support it, you will have to implement methods that can do it such that C = A.multiply(B). Before you start the coding, be sure you look up how to do each of the four operations on matrices. Many times the size of the matrices will determine the size of the result or if they can even be operated on together. Start by writing down the rules first before so that you have a step by step approach to each of these functions.

Added Difficulty: Provide a method which will return the inverse of a matrix if one is available.

Title: Movie Store
Difficulty: 4
Description: One of the most basic learning programs is to create a movie store. Create a program which allows the user to add to their inventory a bunch of movie items and information about those movies. Each movie will have a price based on if they are a classic or new release. They will also differ on the format DVD or Blu-Ray. If it is a classic, the price will be $2.25 and if it is a new release it will be $4.95. If it is also Blu-Ray, add an additional $1.50 to the price. Allow the user to then check out a movie and specify a return date. If not returned by the return date, an additional $1.00 is added for each day past.

Tips: The movie is a class. The store is a class. The store manages movie objects (adding new movies, retiring worn out movies) and how you do this is up to you. You could create a list of movies using an arraylist or vector or some other generic list. It will be this list of movie objects by which the user can select from. Once a movie is checked out, it is unavailable and a return

date is put on it. When it is returned you compare the current date to the return date and if it is past, find out how many days it has been since expiring. Take the number of expired days and multiply it by $1.00. That will tell you how much the user owes in late fees.

Added Difficulty: Create a user account that will keep track of their late fees. If they have a late fee, prevent them from renting a movie until they pay it. Also you could add movie pictures to the movie class. You could also build into the managing class statistical information like the total worth of the inventory and how many copies of each movie the store has on hand.

Title: Patient / Doctor Scheduler
Difficulty: 6
Description: Make a program which allows users to schedule appointments with a doctor. The user can enter information about each doctor and their specialties. Then enter in patients. The program will then allow patients to choose a doctor and an appointment time. Keep in mind that a patient may have more than one doctor. One might be a surgeon and another will be a cardiologist. Don't allow two patients to book the same time for the same doctor.

Tips: First tip is to identify your objects so we can make them classes. First you have a doctor object, a patient object but is that it? What about the relationship between the doctor and the patient? Maybe there is an object there that relates the patient to the doctor. An appointment object perhaps? A day may be filled with half hour appointment objects and each appointment can have a doctor(s) and a patient associated with it. Explore the relationship between a patient and doctor at a given time before you code anything up. How do you make sure that a doctor is available or not for a given time slot? Think of loops.

Added Difficulty: Build in a mechanism to keep track of patient history and which doctors they saw and when.

Title: Product Inventory Project
Difficulty: 3
Description: Create a program that manages multiple products such as shirts, pants, coats and other products. Allow the user to add products to inventory, add stock to an existing product, calculate the total value of a product and the total value of the entire inventory. This program is essentially a product shop in a box. It could be a department store, a specialty shop or a grocery store.

Tips: This is typically the first program that new students get when they are learning a language. It is good for practicing the idea of Object Oriented Programming (OOP), inheritance and polymorphism. It starts off with the creation of a "Product" base class and from there you inherit various types of products. Each of these products are then stored in an "Inventory" class. Start first by creating the Product base class. What would be common to all products? A title, description, price perhaps. Make sure you fully flesh out this base class and keep in mind that you will inherit from it. A product called "pants" is a type of product. A "shirt" is also a type of

product. So they must both share the idea of what it means to be a product. Once you have this done, go to fleshing out the Inventory class which will have a list (array, arraylist, vector etc) of Product classes. Since a shirt is a product, it can be stored in this array long with any other type of class that inherits from “Product”. You want to be able to do products[0] = new Shirt(); where the products list is of type “Product”.

Added Difficulty: Add pictures to each of the items and manage stock levels. When a product falls below a certain number, have the system tell the user that they need to restock.

Title: Recipe Creator and Manager
Difficulty: 5
Description: Develop an application that manages recipes. Recipes can be kept in a manager and divided into groups based on their type. For instance you may have recipes for desserts, beef, pasta, chicken, lunch, salads etc. Allow the user to create their own recipe using a list of predefined ingredients. Have the user create a recipe, specify which ingredients to add from a list and then enter the proportions. For instance, if they are building a salad, they create a new recipe, put it into the salads category, specify romaine lettuce (chopped) as an ingredient and then specify 3 cups.

Tips: Recipe is an object obviously. A recipe will contain ingredients. Each ingredient will be another object. You can add these ingredient objects to the recipe so obviously the recipe object will have some sort of list to hold all the ingredient objects. Perhaps “Ingredient” will be a base class and one of its members will be “proportion” along with the unit type like “cups”. Create an instance of “Romaine Lettuce” which is a type of Ingredient, set its proportion to 3 and its units to cups and then add it to the recipe object. The recipe class is in charge of managing the list of ingredients as well as information about the recipe in general (name, description, type etc).

Added Difficulty: Build in a saving and reloading mechanism which will allow the user to open up your application and fetch a recipe that they saved. You can use a database or file to do this.

Title: Shape Area and Perimeter Classes
Difficulty: 3
Description: Create a set of classes that represent a rectangle, triangle and circle. Have these classes inherit from a base class called “Shape” and each of them will implement at least two methods. One called area(), which will return the area of the shape, and another called “perimeter()” which will return the perimeter of the shape.

Tips: The base class “Shape” will have the two methods area() and perimeter() but they will be empty. They are designed to be overridden by inherited shapes. So make sure that any shape that you inherit from the base class implements their own versions of area() and perimeter() based on the type of shape it is. It is suggested you start with a square since this should be the easiest to implement. Create a Shape base class, inherit a square from it and override the two methods. This should give you the idea for the others if you have done things correctly.

Added Difficulty: Can you make the shapes three dimensional objects like a cube, sphere and pyramid and return the area() for them?

Title: Shopping Cart
Difficulty: 7
Description: Design and implement a shopping cart system that allows the user to enter in various products and display them to potential buyers in the store. The buyer can choose an item, place it in their shopping cart and go through checkout to calculate the subtotal, tax and final purchase total.

Tips: Here we have another instance of Products being an object. The shopping cart will also be an object that will hold these product objects. To figure out the subtotals you would loop through the cart, add up the sum (by getting the price from the product object and multiply it by the quantity the buyer wants), calculate the tax based on this total and add it to the subtotal to find the order total. Depending on the language/medium you are using to implement this project, you may have to keep track of a session so that after putting an item in the cart, the user can continue to shop without losing those items. Then when they are ready to buy, the cart will contain all the products. Make sure the cart has the option of removing items and changing the quantity of a given item.

Added Difficulty: Save the contents of the shopping cart between sessions (opening/closing of the program or between visits to a web site). Add in bundling where two items can be bundled together for a discounted price. Here bundles may be another object type which also has a discount property and contains a list of bundled items.

Title: Software Cataloguer
Difficulty: 6
Description: Make a program that allows the user to quickly catalog which type of software is on their system. Not necessarily installed, but could be used to just keep track of where a binary executable may be located, giving it a reasonable name and when it was added. For example you may download a copy of WinAmp but not install it. Perhaps it is an older version. You can have a program which allows the user to add WinAmp as an entry, that it is located in c:\Temp and was put there on June 7, 2011. The program can also let the user know when an item that was suppose to be there is no longer located there (it has been moved or deleted). This item may have “Winamp 5.21 - Basic install - Downloaded this for a friend” as an entry.

Tips: Each entry in this program will be an object and can contain all the information about the specific software item. It could have properties such as the name of the binary file, the name given to it by the user, a small description of what it is and when it was first added. The idea is that after some time the user will have a list of all the software they have downloaded or added to their system in alphabetical order and quickly keep track of any files that they may normally forget about. It can also alert the user if a file moved without their knowledge.

Added Difficulty: Allow the user to sort based on various items such as file size, file location, what type of file it is.

Title: Student Grade Book Application
Difficulty: 4
Description: Write an application that keeps track of grades for a classroom of students. Allow the user (who is probably the teacher) enter in various assignments, their point totals and their student's names. Then allow the user to enter in scores on each of these assignments for each student. The application will then automatically calculate the average score for each student and assign them a letter grade based on that score. Have it also calculate a class average.

Tips: You can make the student a class and keep track of things like their name, address, phone and any special notes. The scores can be either implemented as parallel arrays or in a 2D array which may be wiser if you wish to use a simple loop to calculate averages for the student; and class as a whole. Student "Tom" can be stored in Students[0] and Scores[0][0] can be his score for the first assignment and Scores[0][1] can be the score of his second assignment. The class score will then be the sum of all students final scores going down the scores array.

Added Difficulty: Allow the teacher to mark the lowest two scores as "droppable" meaning that they won't be calculated into the student's overall average. A teacher may also want to add weights for various assignments. One assignment may be 50 points but makes up 40% of their overall grade for the class.

Title: Vending Machine
Difficulty: 5
Description: Develop a program which acts like a soda pop vending machine. The user must first enter in some money, select an item from a menu, the machine will remove 1 from that pop item and return the correct change. Build a menu to represent the items in the machine and their price. Ask the user to choose 1 item, then enter in a dollar value. It prints "Vending machine dispenses item ____ and the change is _____"

Tips: Each item in the vending machine can be a class of "VendingMachineItem" with a name and price along with the quantity on hand. When the user selects one of the items, you can use that object to get the price and check to make sure there is sufficient quantity to sell. If there is, and the money they supplied is enough to purchase the item, you can remove one from the quantity and subtract the price from the amount the user entered. Figure out the change and then display the message to the user.

Added Difficulty: If the quantity of a product reaches below a certain value, have it write a log saying that the product is low for the vending machine company to review. If you would like, you

can have the program communicate to a website or a database. Also real machines don't usually dispense bills. So have the machine calculate the correct coins to dispense the change.

Threading

Title: Bulk Thumbnail Creator
Difficulty: 5
Description: Develop an application that allows the user to specify a source folder of images, a destination folder and integers for the width/height (or percentage). It will go through each image in the source folder, copy it to the destination folder and resize it to fit the specified dimensions. The result is that the destination folder will contain a list of thumbnails of the images in the source folder.

Tips: Image manipulation can be a resource intensive task. Especially if you are manipulating multiple images that order into the thousands. We don't want this task to lock up the user interface at all and allow the user to still do other things while the thumbnails are being created in the background. So start by collecting the information from the user and once collected it will start a thread. This thread will be passed the function where it will then do the copying and resizing. When it is complete the thread will stop and at that point you can let the user know the task is complete. Keep in mind that some scripting languages may need timeout limits adjusted to compensate for the processing time on large image manipulations.

Added Difficulty: Try to build in a progress bar that will reside on the user interface. For the background thread to communicate with this progress bar you might want to think about using a delegate. It will let the use know how much time is left in the processing of a folder.

Title: Chat Application (Remoting Style)
Difficulty: 6
Description: Make a program that will allow a user to communicate with friends and family. Instead of making one that is of an IRC style, try creating a chat application that is more in line with a personal messenger. Allow the user to send messages to another user and receive messages as well as set their status which the other user can see.

Tips: Start by writing up a communication model on paper. How would the two users find one another? How would they establish a connection and know when one another is ready? How would you send a message and the other receive the message? Are you going to use an existing protocol or one you create yourself? Once you have this mapped out first, you will have a better idea of what you need. Next is to create a demo of two clients which you can chat back and forth with on the same computer. When you receive a connection from another client, move it to a thread which you can send data to and wait for data to come in on. Think about using a buffer to temporarily hold message fragments until the entire message comes in, then dump it to the appropriate user connection.

Added Difficulty: Once you are able to show messages back and forth and get the presence working, can you send a file?

Title: Create a Progress Bar for Download
Difficulty: 3
Description: Make a component that allows a user to add a progress bar to their application. This progress bar can be notified by a thread to let it know the progress of a particular activity. Allow the user to set its size and placement on a form and show the percentage.

Tips: The progress bar is a GUI element and would go on the GUI thread. Meanwhile, the thread that would power it is most likely separate. This will allow the user to continue working with their application while the progress bar continues to show progress through a task. One way to do this is through a delegate which will point to a function in charge of updating the progress bar. You might also look into other designs which involve listeners and events.

Added Difficulty: Change the color of the progress bar as it continues towards 100% if supported.

Title: Download Manager
Difficulty: 6
Description: Write a program that allows the user to manage and schedule multiple downloads simultaneously from the web. The user would enter in the URL of a resource they would like to download and possibly when. The program would then download the resource in parallel with other downloads they may have scheduled. Make sure the program allows the user to set how many downloads can be done at the same time (for bandwidth reasons) and add more downloads while others are in the middle of downloading. The user should also be asked where they would like all the downloads to be placed.

Tips: Each download is going to spin off its own thread. The threads themselves can be very similar but the download manager will need to manage each thread while it exists. When the download is done, errors out, times out etc the thread will close and cease to exist. The application can take the URL, reach out to the server, if it responds with the appropriate connection, accept the connection and put it on its own thread. Then start the thread to download the file.

Added Difficulty: Allow the user to be able to pause and restart a download if the server supports it.

Title: News Aggregator
Difficulty: 4
Description: Create a solution where the user is asked to input several news sources (XML, RSS, possibly a web page, stock quotes, CSV files etc) and the program will, at regular

intervals, reach out and pull down the latest information to display to the user in a clean interface feed.

Tips: Depending on the type of source they are connecting to, you would create a thread for each source and it would regularly check for content. It then will communicate with the “main thread” to update the user interface with the content it has retrieved. If the source is an XML feed, it may need to be parsed to pull out the data... same with RSS. If it is a web page it may need to be scraped. If it is a CSV file it may need to be split up etc. Before you design those pieces however, make sure you create a specification each item must conform to and that your program can work with despite the underlying source. Perhaps this is a news item object that each thread has to create and fill up with data. Then your application can have a list of these news item objects and do things like sort or categorize them.

Added Difficulty: Create tabs which each news source type that can be added to. Perhaps one tab is all RSS feeds while another is page scraped material.

Title: XML Sitemap Generator
Difficulty: 4
Description: Make a program that can crawl a web site and build a XML site map. These sitemaps can be either given to search engines like Google or stored in a database for easy reference. Perhaps the sitemap can crawl only parts of a site and list all attached resources and validate their links.

Tips: We know that Google already has a service that does this, but perhaps you want to create a tool that allows the user to create a sitemap in different formats. Maybe it can crawl the same site and then generate one sitemap for Google, another for Yahoo and another for Bing. Maybe it can be a sitemap that is then sent to another application for further processing. The first step is to simply come up with the format. If it is for Google, read their site map specification. If it is another system, see what that system needs in the way of a format. Then have your crawler (which is on multiple threads to reach different pages at the same time) build the pages into this format and either present to the user or submit to sites for them.

Added Difficulty: Allow the user to choose which formats to put the site map into and support at least 3 different formats.

Web

Title: Bandwidth Monitor
Difficulty: 7
Description: Design a program which allows the user to monitor how much bandwidth they are uploading and downloading through their Internet connection. Have it show the number of packets up and down and on average how much they are sending/receiving per second.

Tips: Some languages have classes that provide network information and properties for bytesReceived vs bytesSent. In C# for instance this is done through the System.Net.NetworkInformation classes. It is pretty difficult to get the entire bandwidth usage for the given machine but we should be able to track what comes through an application or ask the operating system to track that information for us (like we have done in C#).

Added Difficulty: Provide different levels of measuring from average per second to average per minute to number of bytes and perhaps top speed reached.

Title: Bookmark Collector and Sorter
Difficulty: 6
Description: Develop an application that can give the user the ability to bookmark sites, stash them away, sort them and share them across the web. Provide a mechanism for others to see the bookmarks and visit them as well as rate each one with a thumbs up or down. Keep stats on how many people have thumbed up/down the bookmarks for a given user's collection. Provide a mechanism for grouping bookmarks into categories of the user's choosing.

Tips: The best place to start with a project like this is to write down the steps of getting a bookmark off a user's browser to the place where the collection will be stored. Depending on your language here, you may opt to create a tie in to a context menu on the browser, read an existing bookmark file or something in between. Once you have the source of the bookmarks, how will they be stored? Think about providing a bookmark class which can tie up all the information about a bookmark (including how to compare bookmarks for a sort). For the grouping functionality, a group class can contain a list of bookmark class instances that can then be added to, edited and removed. This group class will also be the one responsible for doing the sorting and sharing. The sharing can be done by simply opening up a connection to a web server, logging in and storing the bookmarks as a file or inserted into a remote database.

Added Difficulty: Record the page's favorite icon (if it has one) along with the bookmark.

Title: CAPTCHA Maker
Difficulty: 6
Description: Make a program that allows the user to specify various length strings which are then presented to the user in a CAPTCHA widget. This text should be easily read by a human being, but unable to be deciphered by a computer using pattern recognition. These types of widgets are used for form validation to make sure that the person filling out the form is a human and not an automated program.

Tips: This program is going to require you to break out some knowledge on graphic drawing dynamically. The image which is presented to the user must first read a random string from the list of predefined strings and apply a distortion or obfuscation to it just enough to make it easily readable to the user (where they can then enter reply with the data) yet not be recognized by

computers which apply pattern recognition that identifies letters and numbers. In PHP you can get access to the functions you need by making sure you enable the GD drawing library functions. Think about how you want to write the letters/numbers including its tilt, if you want to draw lines or dots over the lettering and where each letter may be placed on the graphic.

Added Difficulty: Apply a distortion that warps the letters using an algorithm of your choice.

Title: Content Management System
Difficulty: 8
Description: Content management systems are most certainly useful with some of the most popular being Joomla, Drupal, Sharepoint etc. Create a program which allows users to login, create content, publishes content and manages that content using folders, groups and permissions. Provide a mechanism for uploading content, limiting which users can see a piece of content and edit/approve articles.

Tips: Content management systems can be simple or extremely complex. They can also be wide ranging in format and goals. First identify which goals you want to achieve with the system. Is this system going to be better suited for a particular company or industry? Next step is to identify the classes that you would use in such a system. Some classes may be: User, Group, File, Role, Video, Image, Upload, View etc. Model out each of these classes and try to identify how each relate to one another. How does a user instance fit in with a group class instance? How does a group have access to one file and not another? This will flesh out your relationships between the classes. These classes and relationships will also form the foundation of a database design. Keep in mind that a database design should be similar to the programming code design. Start small and grow from the beginning. Always design with the idea of a login and permissions. Always keep in focus what the business needs are as well. Just because you can put a permission on an individual file it may not be what businesses are looking to do in their industry.

Added Difficulty: Provide versioning of content.

Title: Countdown Screen
Difficulty: 2
Description: Make a web page which features a countdown timer on it. This countdown will be counting down to a specified date and time entered by the user. These types of screens are often seen when it comes to a website being launched. When the deadline is reached, have the timer stop and display a welcome message.

Tips: One of the easiest ways to do this is to use JavaScript to setup a time. The user would enter in a date and time (the launch date) and the script will take the date now, convert it to a timestamp and subtract that from the timestamp of the deadline. This will typically give you the number of seconds left in the count down. All that is left is to convert the number of seconds to minutes/hours/days format.

Added Difficulty: Take into account a user's time zone and see if you can display the clock using graphics instead of straight up text.

Title: E-card Generator
Difficulty: 4
Description: Develop a program or site that would allow a user to generate electronic greeting cards using images, possibly some flash, specify a message with a certain font and a sender to send it to. The user would then be able to send the e-card to that sender.

Tips: This project is pretty straight forward. First gather your resources by picking up some imagery, and perhaps some lovely poetry, and create an interface that will allow your user to easily pick between them. Create a template file that shows how the card looks as the user builds it and add some drag and drop functionality to either drop the graphics on the card (using something like JavaScript... maybe jQuery) and click to add text. You could also go another route by showing the card with images already in place and just ask them to pick which card and where they want to write the text. Make sure you pick images that are not copyrighted and free for commercial use. Perhaps you can purchase some images off of iStockphoto as long as the license allows it.

Added Difficulty: Allow the user to insert video and audio clips into their cards.

Title: File Downloader
Difficulty: 5
Description: Make a program that allows a user to quickly download all files on a web page of their choice. If they go to a listing of files, the user would be able to select a menu option or run their program and it would scrape the links, queue them up for download and then start downloading one or multiple files at the same time. Let the user also specify where the files will download to on their local machine.

Tips: This project is all about identifying the links on the page (try using a regular expression) and then establishing a connection to each file, requesting it from the server and then saving it locally. Try looking at threads to help with simultaneous downloading of multiple files at the same time. Start with trying to get the program to download the first link it runs across and once you have that working on its own thread, it should be a bit easier to expand it to include multiple file downloading.

Added Difficulty: Show the current status of all files being downloaded and once all downloads are done, generate a report for the user to see if any file downloads failed.

Title: Media Player For iGoogle
Difficulty: 6

Description: Implement a widget on Google's widget search page iGoogle that allows users to either choose a file from their computer to play (a MP3, Wav or video file) or have it stream an audio clip from a specific server and play it through the web page. The goal here is to have a widget that someone can add to their iGoogle page that allows them to play music no matter where they are.

Tips: First stop, Google's iGoogle Developer Homepage and learn how widgets work. Start with something simple if you have not developed a gadget before. Some of the easiest gadgets to make are HTML based ones so look at possibly putting a player on an HTML page and then integrating that code into the widget. The player itself could be flash based, a Java applet or perhaps something done with HTML5.

Added Difficulty: Build in playlists so that the user can build their own list of favorite music and play that list whenever they want.

Title: Music Player and Manager
Difficulty: 6
Description: Allow the user to manage their music library without having to do it through a program like iTunes and do it from the web. The manager should allow the user to add, delete, rename, sort and alter a song's meta data. It should also allow the user to select a file and play that file as well as pause, skip, stop and start music.

Tips: This project can be divided into two pieces. One piece is the actual player of music and the other is the manager. It would be advisable to first get the player working and then provide a way you can easily feed music files to it. If you are doing it in Java then you can look to the Java Media Framework API (JMF) to help you out. For the web, you might want to grab an existing Flash player or create your own if you are doing this project through Flash. Once you have the player working, you can create a menu item which launches the manager which will manage the files. Double clicking a sound file in the manager can then be fed to the player. The player itself doesn't have to know anything about the manager to function so if you find yourself trying to get information from the manager about files, then take a second look at your design.

Added Difficulty: Add in social networking functionality so that when you play a song it lets your friends know on Twitter the song name and artist.

Title: Online IDE
Difficulty: 9
Description: Create a website that allows users to write in some code into an online editor and then run that code. The code it runs can be based on your own language or one that already exists.

Tips: This can be a difficult assignment because it essentially makes you write a compiler or create a web interface to an existing compiler. You then have to pass the code from the user to

the compiler/interpreter and return the results. Not only that, but you have to keep everything in check so that the user can't maliciously write code that will affect the web server. So start with checking out a few sites that already do this like ideaone.com. There are ones for Python, Ruby and even Java out there. Second, think about how you will put code into a sandbox type of environment where the code someone writes can't reach the server to do harm. Lastly, how would you pass the data to the back end and then back to the page to show the user the results?

Added Difficulty: Support more than one language.

Title: Online Whiteboard
Difficulty: 8
Description: Make a program online where multiple users can interact with one another using a web page drawing surface. One user can use the mouse to draw or type a message on a page and the user on the other end can view the drawing and draw back. This can help facilitate doodling out ideas as they talk through some other medium like the phone.

Tips: JavaScript, and in particular AJAX, is going to be your friend here. That or use HTML5 in combination with JavaScript and AJAX. Start by capturing mouse events for a given item (like a div tag) and based on the event have it draw onto that element (the HTML5 canvas element is great for this). As the drawing takes place, you will make calls through AJAX to a script which can pass drawing data to other users on the page. How would it know who the other users are? When they (user X) loads the page, it sends a request to a server-side script or database and "registers" them as someone receiving drawing events from user Y. Every time user Y draws, it looks up those registered for that user and sends the data to them. You may also need to keep a drawing state on the server that you can share with newly joined members. That way they can see what has been drawn before they got there.

Added Difficulty: Make a picture tube type of tool that allows users to choose from specific pictures and stamp it onto the page. The user on the other side can then see those stamps.

Title: Page Scraper
Difficulty: 6
Description: Design a program which allows a user to quickly visit a page and scrape off a particular element from it like links, images, div tags etc. Prompt the user to enter the type of tag, or a pattern to look for, and it will request the page and return all matching elements.

Tips: This is a project which requires two things. One is to request the source of the page so we can have some text to search. The second is to build a pattern using a regular expression that will then search that page source and return all matches. If you are unfamiliar with regular expressions, you can get started with a site like Regular Expression Tutorial that shows you the basics and build on from there. Keep in mind that HTML source is full of matching tags so you

may want to look for things like <a> and . You may also need to take a look into "less greedy" pattern matching. (In other words things like .*?).

Added Difficulty: Output all matches to a file or database for later review, sorting or indexing.

Title: Password Safe
Difficulty: 6
Description: Develop an application that allows a user to store away their passwords. The user is prompted for a username and super password which then logs them into their account where they then can create, edit and delete username/password entries for various sites. So an entry should at least consist of the website address, a username and password to get into that site. When storing the information, the username and pass are encrypted and take every precaution to make sure that transmission of data is across a secure SSL connection.

Tips: This project has a huge cost if the data was to ever get out so start by first creating the design of the site and how you can manipulate data through a secure means. Avoid using a hashing algorithm like MD5 and instead go for a tougher one like SHA-256 or an AES based cipher for storing. Hashes won't help you hear, just encryption because you want to be able to decrypt the information for the legitimate user. Hashes are only one way! Be sure to also obtain a SSL certificate for your server and that it is enabled for any requests and responses to password data. Also make sure no SQL Injection attacks can take place on any forms. You wouldn't want to have the database compromised.

Added Difficulty: Provide mechanisms for searching for entries as well as sorting based on website name. A user may have multiple accounts for the same site.

Title: Scheduled Auto Login and Action
Difficulty: 6
Description: Write a program or script that creates an automated login to a website and perform an action. This automated process could be something like logging into a Facebook account and post to a user's wall or login to a webmail account and check for messages. Collect the necessary credentials and a scheduled interval from the user. Have the process perform the actions at the scheduled time(s).

Tips: The process will differ depending on the site and the process you want to achieve. First step through the process yourself and note all the pages you visit, the information you have to submit and where you have to submit them. For instance, a webmail page may require you go to http://webmail.somehost.com and put a username in a textbox with the name "usr" and a password into another textbox with the name "pswd". Record all these items and how they are submitted. Perhaps the form uses the POST method to submit its data to another script? Once you know the process, automate it in a script that will perform the necessary actions.

Added Difficulty: Look for a site that requires SSL and see if you can get the script to communicate using the credentials.

Title: SEO Optimizer
Difficulty: 5
Description: Put together a script that will ask the user for a list of website pages, and when they are done entering the list, it will visit each of those pages and look for ways to optimize it. It may list recommendations to the user like adding titles to pages, come up with keywords to use, develop a keyword rich description, look for ways to streamline the code or look for missing alt tags etc. After the program is done executing produce a report for the user showing the recommendations for each page.

Tips: First pick out what features of a web page you will advise the user on like missing tags, alt attributes or suggesting keywords. Develop those page analyzing features first. Then once you have the algorithms in place, you will be in a better position to create a program which will establish a connection to each page, fetch the source of those pages and then run them through your algorithms. As with any program that may make multiple connections at once, it would be wise to put these connections on threads so that while one connection is waiting for a response from a host, the others can be connecting to other pages.

Added Difficulty: Build in status bars for each thread to show the user where in the analyzing process the program is on each page.

Title: Shopping App
Difficulty: 8
Description: Make a program that will allow a user to enter in an item by product name and it will bring back the current price multiple vendors are selling it for on other websites. This way the user can then quickly scan who has the product, what they are selling it for and compare matches.

Tips: This is another application which will contact multiple sites and essentially page scrape them for the necessary data. This may also require your program to identify search boxes on those sites, submit a search and pull the search results back to the user's program. How this is done will vary on a site by site basis, so make sure that you either provide a list of well known, non changing vendor sites or interface with any APIs they may have which can get you the requested data. You may also have to rank matches according to the user's query and sort them by rank so that the matches which may meet the user's request the most are displayed first.

Added Difficulty: Try to include pictures or allow the user to enter their own sites as long as they provide key data on where to find pricing and product information.

Title: Telnet Application

Difficulty: 8
Description: Create a telnet program that will allow the user to connect to remote computers, enter commands, navigate systems and communicate with others. The program should provide an interface that allows the user to enter in text which will be sent to the remote computer as well as the replies from that computer.

Tips: Telnet is like many other applications where you first need to establish a connection with a computer and then follow the protocol of communication to interact with it. So first look into establishing a socket connection to a specific IP address. Then following the Telnet protocol (outlined in RFC 854 and similar) write commands given by the user to the given socket. To handle multiple connections you should look at implementing a multi-threaded system where connections can be run on their own connection and managed collectively. This could be done using a list data structure and communication can be done through delegates.

Added Difficulty: Try implementing an encryption method that will make your Telnet sessions more secure. Also try implementing a file transfer mechanism that will allow the two computers to exchange files.

Title: Template Maker
Difficulty: 7
Description: Write an application which allows the user to choose from various options (like programming language type, whether to include a main function, inherit from a certain class etc) and have it generate a “skeleton” template for a program or web page.

Tips: Content generation is fairly easy as long as you know the structure of the specified language or features requested by the user. For instance, if the user chooses to have the program generate a main function and also specify this is for Java, then the skeleton will include a method with the signature “public static void main(String args[])”. As the options get more complex, so too does the generated code. Don’t make the generation too specific otherwise you may alienate your user and overload them with too many options; causing confusion. Start with one programming language and build a program which can generate reliable error free skeleton “stub” code and then branch out into other languages if you choose.

Added Difficulty: Have the program launch the appropriate helper program and load the template into it. For Java this may be an IDE like Eclipse or if it is C# have it pass the template to Visual Studio.

Title: Text Based Game Like Utopia
Difficulty: 6
Description: Make a text based game like the popular game “Utopia” that can be played online. This game requires users to create an online account where they start off in a group or clan, given some land to build upon, and some limited resources to then start constructing an empire. As time progresses, the player makes money, acquires resources, wages wars or establishes

peace treaties with other players. Each hour the player is given a number of “turns” in which they can do various actions. Once their turns are used up, they can’t do anything else until the next hour (which is the next day in the game).

Tips: These games feature three distinct parts. One is the database which keeps track of all player resources, turns, money, number of units, how much land they have etc. Start with a solid design of how all these various assets will work and relate to one another as various tables. The second piece is the progression of time. This is usually done with a scheduled task. In *nix environments this is accomplished by using a task scheduler known as a “Cron job”. You can setup these task schedulers to run scripts at given intervals or at specific times of day. If you have it run a script which looks at player’s accounts and updates their resources accordingly once per hour, you can simulate the passage of a day in the game. The third part is the individual pages that the user interacts with when they buy units, spend money, wage a war etc. These pages feature server-side code (PHP, ASP.NET, Cold Fusion etc) which runs through game logic. For instance, you can’t buy any more units if your money is zero or don’t have enough wood. After designing the database, work on the code pages, think about your script(s) you will run through cron jobs and do those last when assets are ready to go into motion.

Added Difficulty: Try to incorporate a medium like Flash to help animate the process of waging a war or when the user does a specific action.

Title: Web Browser with Tabs
Difficulty: 9
Description: Create a fully functioning web browser which features multi-tab capabilities, renders HTML/XML/JavaScript and can display web page elements comparable to other modern day web browsers.

Tips: One of the easiest ways to tackle a project like this is to identify some ready made web browser controls that might be available for your given programming language. Putting these controls into your project will help you get most of the rendering of pages out of the way and allow you to focus on the all the “other stuff” that is around the control... like tabs, URL boxes, bookmarks etc. One warning with this however is that these controls may depend on various technologies to be installed on the computer like a browser engine. Visual studio’s web browser control renders much like Internet Explorer and have many of its quirks. It may also then expect the user to have specific DLLs installed. Keep this in mind and start testing early betas with various different machines. Get some friends to try it out on a slew of different platforms before putting too much work into the project.

Added Difficulty: Build in a fully functioning options screen that lets users set a homepage, how new tabs are created and when and if they wish to see popups or not (aka a popup blocker).

Title: WYSIWYG (What you see is what you get) Editor

Difficulty: 8
Description: Write an editor control which the user can put into a web page or program and gain the ability of a “Word like” page editing interface. Since many users might not be coders, the control will allow them to enter in text, images and tables more quickly and simply. For example they can highlight text, select a color from a swatch and turn it red with just a few clicks. Perhaps they want to bold italicize some text or quickly drop in a graphic without knowing a single thing about HTML.

Tips: These types of editors have two main parts. They have the HTML part which is done behind the scenes and a rendering part usually made up of a browser window or advanced rich text control. These editors usually let the user flip between the two views making it easy for the advanced user to work with the HTML source and the beginner to work with a more visual representation of the content. One suggestion to help manage these two views is to keep a copy of the HTML source in a hidden control and load it into the editor as needed. When it comes time to render the code for the visual part of the editor, you can read the content out of the hidden control and display it. Whenever a user edits the visual or HTML version of the source, a copy is also updated in the hidden control. Examples of these type of editors include the ones used on Wordpress’ edit post screen and Dreamweaver.

Added Difficulty: Build in various dialogs to help the user pick colors, save the content to a file, show a full screen view and easily set up various fonts and font styles.

Files and Streams

Title: Add Transactions In File and Find Averages
Difficulty: 3
Description: Make a program which asks the user for a file name containing input values (scores, random integers etc) and reads in those values to find the average. If there are 5 values on a row in the file, read in those five values and find their average for that row. Print the averages to screen.

Tips: Well since we are reading a file, we first need to create a mechanism of reading the file using a stream. In .NET you would use something like a FileReader from the System.IO namespace and in Java you may use a BufferedReader or similar from the java.io package. PHP would use a function like fopen(). Focus on reading in the entire line first and then break it apart into its individual values. Keep in mind that averaging is simply adding up the values and dividing by the number of values. You may also need to focus on conversion. You read a number 2 from the file, but is it read as a string or an integer?

Added Difficulty: Have the output of the program go back into a file. After finding each average for a row, can you find an average for all rows?

Title: Bar Code Reader and Generator
Difficulty: 8
Description: Bar codes are those vertical lines found on the back of many products you buy at the store. They are scanned by the merchant to identify, and price out, the item being purchased. Design an application which can read in a bar code using a bar code scanner or generate a bar code which can then be printed for a product.

Tips: There are a couple ways of doing such a bar code reader or generator. Let's start by first getting the program to read a bar code. To do this, you will need an input device capable of reading the bars. You can use a bar code reading gun or one of those light pens that can scan across the bars and translate those bars into the universal purchase code (UPC) which the item is categorized in the system. Once you have that working, you can generate codes and have it read back to see if it returns the values that you encoded into the bar code. Bar code generation can be done using a bar code font that translates a "w" into the appropriate width bar. These types of fonts are usually available for purchase (very rarely free). Another option is to come up with your own bar coding scheme where your program will read "w" and say I need to draw a bar that is "x" pixels wide.

Added Difficulty: Tie this to a database of items which have their UPC values stored. When an item is scanned by your program's input device, it looks it up in the database and adds the appropriate item to their order.

Title: Bulk Renamer and Organizer
Difficulty: 4
Description: Develop a program which asks the user for a source folder name, a file name pattern and various other options used in the organization of files. The program will rename each file in the source file, according to the pattern entered, and organize them based on the options specified. For instance, if the user enters "C:\temp" which contains 100 image files and gives the pattern "image*.gif" it will go through each of the 100 images and rename them to "image1.gif", "image2.gif", "image3.gif" etc. If the user specifies that each of them should be placed in their own folder, it will also create folders "image1" and place "image1.gif" into it, create another folder named "image2" and place "image2.gif" into it etc.

Tips: This program makes use of various functions that read the contents of a directory and also functions that are used in renaming, moving files and creating directories. So start by first looking up which functions those might be for your given language. The next step will be to create a central loop that can loop through the source directory, find a specific file, work on manipulating it and then move onto the next. This process will continue until the directory has no more original files. One thing to keep in mind is that a file may be locked by the system. So be prepared to encounter this and catch any errors that might be generated. If a file is locked, move onto the next.

Added Difficulty: Break this process out onto a separate thread for those directories which have a large number of files. Keep a running progress bar of estimated completion and general stats like how many files are processed, the size of all the files processed so far and how many were not able to be worked on due to being locked/inaccessible.

Title: CD Label Maker
Difficulty: 3
Description: Create an application that allows the user to create and print CD labels for their CD collection. Allow the user to enter different fonts, graphics, colors etc and then print them out. Provide a print preview so they can get an idea of what the finished label would look like.

Tips: Start by measuring a CD's diameter. Next you can create a drawing surface in your program which is square and has the width and height equal to the CD's diameter. This will be the label and when you draw within this surface, imagine a big donut shape within it. In other words, the exact middle of the square is not going to show because that is the hole in the CD. Likewise on the four corners where the CD won't reach. You can allow the user to draw into these areas but let them know that they will eventually be cut off. All that will be left is that when they go to print, make sure that the square you print on the page is lined up with their label stickers. You may want to buy a pack of blank labels to experiment on.

Added Difficulty: Have the program shade out the parts of the square where the label will not be printed. You can do this by drawing a circle on the square and shading everything that is outside of this circle.

Title: Clipboard Manager
Difficulty: 6
Description: Develop a program which makes it easy for the user to hold various items in the clipboard at the same time, switch between them and paste from the clipboard into other applications. If they copy three items in a row, add those three items to the manager's clipboard, then allow them to select which they want to paste.

Tips: Copying information to the clipboard should be a binary process. This is because they may not be just copying text, they could be copying files, images, video etc. You may also need to look at handling the ctrl+c and ctrl+v key combinations from the OS and trigger your app to read the clipboard and store values in a dynamic list structure of binary objects. How you accomplish the actual storing of this data is up to you, but look at how you would store such a binary object and think about how to store it first before you actually code your application. Especially since this part of the program is the heart of the manager. Pasting will be simply putting that data back into the clipboard and then allow the user to paste from there.

Added Difficulty: Have your program show thumbnails for images stored and various icons for other file types stored.

Title: Code Snippet Manager
Difficulty: 5
Description: Write up a program which allows the user to quickly and easily store code snippets, functions and other code text for later use. When a user comes across some code they want to store, they highlight the text, click a hotkey and bring up the program to then paste it into. The program should have the ability to save text, organize text into arbitrary groups and fetch/browse the stored code. Store this code either through a database or through a file.

Tips: The core piece of this program is a text control which can take in text or show text to the user. Then by building some saving code, it can create a code "item" which is then either stored in a database or text file. So first it is recommended to create a class which can store everything about a code item. Perhaps it has an ID, name, date it was created and maybe a parent ID to associate the snippet with a group. Then create a form where you put on the text control. When the user hits save, it reads the code in the textbox, determines if this is a new item or an existing item. If it is a new item, create an instance of your item class and set its properties. If this is an existing item, fetch that class, then alter it through its properties. Saving will then simply write the values in the class out to disk or execute an SQL insert/update.

Added Difficulty: Try to hook the program into the user's preferred browser. If they highlight some text, right click and then see a menu option to store the text, that would be preferable.

Title: Color Picker and Translator
Difficulty: 4
Description: Make a small tool that allows the user to quickly pick colors from a color swatch. It will then return the Hex and RGB color values for that color. It should also provide a place where the user can enter in a Hex value and it will convert it to RGB and vice versa.

Tips: The main screen should display the color swatch where the colors can be generated using a mathematical formula. You can find such formulas as www.easyrgb.com and when they select a color, you can quickly break the color down into its RGB and Hex color values. It is suggested that you make a function for each of the color value types like convertRGB() and takes a Hex or color class object as a parameter. The output would then be a string or an array of values.

Added Difficulty: Include CMYK conversion and perhaps build in the option to pick complimentary colors for use in themes. For an example of this, check out kuler.adobe.com.

Title: Create Zip File Maker
Difficulty: 7
Description: Write a program that allows a user to select multiple files and zip them together into a zip file archive. Provide a way for the user to choose the different levels of compression and the ability to view files inside a zip file. Also give the user the option to extract the zipped file(s) and specify a destination to extract the files to.

Tips: Depending on the language, zip file creation can be an easy or difficult task. Start by first finding the functions or extensions needed to handle zip archives before attempting to code anything. Java has the java.util.zip package while the .NET languages may need an archive extension like DotNetZip to give you the ease and functionality for handling zip files. Python has the zipfile module that may life a little easier for those programmers. PHP programmers have the zip functions too. Once you have those functions identified, you will be off to a good start in creating an interface that quickly allows the user to select files, build zips, then view those zips and extract. Your testing should include this process of building, viewing and then extracting files from zip archives.

Added Difficulty: Support more file formats including GZip, 7-Zip and Rar files.

Title: Envelope / Label Printer
Difficulty: 6
Description: Create a program that asks the user for text and to select a given letter/envelope size (from a menu). It then prints the text to the appropriate dimensions to fit on the letter, label or envelope.

Tips: The trick here is the printing. Keep in mind that printing in pixels is different than printing in measurements such as inches/cm/mm etc. Be sure that when printing you take into account a font's height, spacing and other metrics. Also keep in mind that not all fonts on a computer are easily readable when printed. Look to serif fonts such as Times New Roman which typically look better when printed than do fonts which are designed more for the web (like Arial). One great site for learning about a font's metrics is the MSDN Font Metrics page which talks about ascent/descent and line printing. If you are going for a visual look to the program, you may want to ask the user for the letter type then create a screen which shows the letter/envelope to scale and allow them to draw text on top of it. If this is the plan, look to the GDI to draw text on top of the graphics.

Added Difficulty: Support more letter types and provide padding/margin support. Can you design the program to post a return address exactly in the upper left corner within a 2 inch by 1.5 inch rectangle?

Title: Excel Spreadsheet Exporter
Difficulty: 6
Description: Design an application which asks the user for a given excel spreadsheet (or any type of spreadsheet really) and exports the data into a comma/tab delimited formatted text file or SQL file. The file should be in a format that can then be directly imported into a database. Give the user the option to browse for a file on their system, see a preview of how the content will look exported and then make adjustments before exporting.

Tips: Accessing Excel spreadsheets may require you to go through the use of an Excel class or library already developed by Microsoft. In .NET these libraries are usually included in the

“Choose Items..” menu option when you right click the toolbox. They will be located on the “COM” tab as the “Microsoft Excel Object Library” and may only be present if you have a copy of Microsoft Excel installed. If you do not use Microsoft Excel products, you may need to find generic code for accessing spreadsheet data to base your code off of. Once you are able to read this data, it is just a matter of formatting the text and writing it back out to file. Beware of certain characters, such as double quotes, which may interfere with parsing of spreadsheet data or may need to be added to the spreadsheet data prior to writing. SQL files may need to be written as SQL statements that insert spreadsheet columns as columns in a database using INSERT syntax.

Added Difficulty: Once you have your program exporting, can you make it also convert to other formats from Excel? Can you export spreadsheet data as HTML tables or PDF?

Title: File Copy Utility
Difficulty: 3
Description: Some users are not happy with the standard select and copy features and want something a little more specialized. Create an application which allows the user to quickly select files based on a specific file pattern and copy those files to multiple locations. For instance, given a folder full of .gif, .jpg and .bmp images, the user can select only .jpg and .bmp images to then copy all .jpg images to the JPEG folder and the .bmp files to the BITMAP folder.

Tips: This program is great for beginners who are wanting to do learn about generalized file system functions. The program should probably have a dialog window that first allows the user to select the folder(s) to work with along with options to do various tasks (eg. Create a folder for each image matching the pattern?). Then loop through those folders reading each file’s properties and comparing them to the criteria specified by the user. This is usually done with a while loop that continues until all files have been read and executes various statements within its body. Be sure to check for files that are actually folder names, quick ways to eliminate the eligibility of a file (filter it out) and avoid intensive function calls.

Added Difficulty: Support recursion into sub folders with the option to limit how deep to go down the hierarchy tree.

Title: File Explorer
Difficulty: 6
Description: This project is to simply recreate the windows explorer program. Design the program to include a treeview structure on the left with several nodes representing drives and folders. On the right is a viewing “pane” that allows the user to view files in a given drive/folder. Support a detail listing view, small icon view and thumbnail view for these files/folders. When right clicking a file/folder, provide the basic options of opening the file in an associated program, deleting the file, renaming the file or setting its properties (hidden, read only etc).

Tips: There are several parts to this project. First you will need to be able to read all the drives on the current computer when constructing the treeview. Look for functions in your language of choice for detecting drives and be sure to check which are fixed, which are removable, which are CD Roms etc. The second issue you will surely run into is displaying folders with tons of sub folders. Think about ways of loading these sub folders dynamically upon opening of the specified folder. Do not try to load all folders at the same time. This will increase performance.

For the viewing pane, the details view will simply read various properties (modified time, created time etc) and put it into various columns. The small icon view will need to read the extension of the file and asking the operating system which icon is associated with that file extension. While you are at it, also ask the OS which program opens that specified file and save it so that you can reference it when it comes time to open the file using menu options. Thumbnail view may be the most difficult because while some files may be videos, and need to fetch a frame from that video to use as the thumbnail, others may not have a viable thumbnail (like an audio file). In the case of not having a thumbnail, use a large default icon for the thumbnail.

Added Difficulty: While in detail list view, build in sorting so that when you click any of the column headers it will sort the files based on that property.

Title: File Splitter/Joiner
Difficulty: 5
Description: Develop an application that allows the user to specify a large file and split it into smaller files. Then allow the user to select these smaller files and join them back together into the original large file. The user should be able to choose how many files to split the larger file into (either by number of files or size of each file) and select where to put the smaller files. Each file should have a sequential number in the file name, along with sharing the same base name, to let the user know how many files they have and which chunks belong together.

Tips: One of the easiest ways to do something like this is to treat each file as a binary file where you read a specific number of bytes from the file, save it to its own file and when it comes to joining them you can locate each of the smaller files and write them all to one large file again. These types of files use to be used quite common back in the earlier days of computing when disks such as floppies were much smaller and couldn't hold a larger program on their own. Read each byte of the larger file and write it to a new file until you reach a chunk limit or the end of the larger file. When joining, read the entirety of a chunk and write it to one file. Thus the beginning of chunk 2 would follow that of chunk 1. Look to binary file reading for more information.

Added Difficulty: Can you develop a system where it then writes the chunks to various disks (CD-Roms or floppies) and be able to read those back from the disks to join them?

Title: Font Loader
Difficulty: 3

Description: Make a program that shows a sampling of all the various fonts on the system quickly and easily. Allow the user to view all the font names they have, rotate through them and view sample text using the font. The user can choose from different font sizes and styles that are supported by that font. Make sure no size or style can be selected for a font that doesn't support it.

Tips: Many languages including .NET and Java have font specific objects which would be of great use for displaying fonts. You may also want to look to libraries such as the FreeType library for helping with font properties (to know which properties a font supports). You can usually do a quick search of a system to locate where the fonts are installed and then you can iterate through them. When you load up your program's main screen, and when the user selects a font, you should fetch the file, create a font object with it, get its properties and specify where you can write to the screen using the font. Keep in mind that some older languages may actually require you to use a Brush or Pen object to specify the color or the width of the line. You can also think about storing these font objects in a list data structure for easy reference as the user flips through multiple fonts quickly.

Added Difficulty: Build in the ability to have your program visit various websites, view their fonts and install them onto the user's computer through the loader. Essentially a view and "shop" for fonts program.

Title: Generate Invoice/Purchase Order
Difficulty: 5
Description: Write a program where the user can create custom made invoices or purchase order forms. The user should be able to use a visual display, like a blank form, and drag and drop headings, labels, draw lines and create an area to display order items. These order items can be created by a record set, table, database query etc. This application is to behave much like Crystal Reports. The template invoice/purchase orders can be created, edited and saved for later use.

Tips: How you go about tackling this will depend on your language and medium of choice. A suggestion would be to try this out on the web first where the template itself can be a file filled in with variables. PHP would be a good choice for this. Where labels go you would have tags (like {{tag}}) which are then filled in using results from a query or stand alone text strings. The order items area can be a standard HTML table and the subtotals/totals can be calculated on the fly as results are iterated through. Once this is done from a web perspective, it would be easier to port over to programming languages given the overall design could be similar. The templates themselves can be saved as stand alone text files.

Added Difficulty: Allow the user to also insert pictures like a company logo for the form's header.

Title: Image Map Generator
Difficulty: 4

Description: Design a program that asks the user for an image and then allows the user to draw "hot spots" on top of it to represent links to other sites. The type of hot spots they could create would be circles, squares or multi-point polygons. For instance, a picture of a street corner could be loaded and the user can click around the street sign to make it a hot spot. Using this image map on the web, the visitor can then click that sign and be taken to a website. Look up "image maps" for more information on how these work.

Tips: If you are unfamiliar with image maps, they are images which have clickable areas (called "hot spots") on the image that can go to different sites or trigger different actions. They do this by keeping track of a map of coordinates in the HTML source code. Each hot spot has a list of x,y coordinates (which form boundaries) and when the mouse enters any of these coordinates, it knows that it is a hot spot and to lookup either a website URL or trigger some JavaScript. Look up "Image Maps" on Wikipedia for more information on the structure of these maps.

Added Difficulty: Allow the user to specify JavaScript functions to be called when a hot spot is clicked.

Title: Log File Maker
Difficulty: 5
Description: Develop an application that allows a user to create various log files. This program can be used for web applications, could be integrated into a control for a form or be called as a DLL. Ask the user which types of fields they want to be in the log file, the file's name and provide an appropriate mechanism which would allow the user to call a write command, passing it those fields to record.

Tips: This is a simple file output program that will keep track of various fields to write. The trick will be the write command which should be able to take in a variable number of arguments (think default parameters) and match up those arguments with the specified fields to create a "record". Then the log file maker would write this record to the file, one per line. For instance, if I specify I want to write the time, user name and what file they requested for a web server application I would be able to write a record using a write command such as logfile.write(time, user, file_path). It could then match up each of those arguments with the appropriate fields to produce a record that looks like [01:23 am] George - c:\some_path\file.html. For writing, be sure to create a function specifically designed to write a string to a file. After you create the string you can then simply give it to this function to write it to file.

Added Difficulty: Allow the user to specify file names based on wildcards. So if they specify logfile_*.log it will create log files based on the date like logfile_20110719.log.

Title: Mp3 Tagger
Difficulty: 6
Description: Create a program where the user opens up their favorite Mp3 and specifies information about that file such as artist's name, year it was released, album it came from, track

number etc. Make it only record ID3 v1 tags to start and allow the user to edit all existing tags that are currently found in the file. After making the edits, the user should be able to verify the changes were made by opening the file backup and all their changes should be read back.

Tips: ID3v1 tagging is found in the first 128 bytes of an Mp3 file. The first three bytes should be read as “TAG”. If this is not found, the file does not have ID3v1 tagging on it and thus needs to be created. After this tag follows several fields each given a set number of bytes of space. You can find all the information you need about these tags at the ID3 entry on Wikipedia. To help facilitate the writing of these tags, consider creating a structure in your program with a set number of bytes that match each field. This way you can pass around the structure between various functions to read, write and save to the file.

Added Difficulty: Once completed with ID3v1, try to go to ID3v1 extended tags and then ID3v2 where the user can enter in picture information among other things.

Title: PDF Generator
Difficulty: 8
Description: Create an application which generates PDF files from other documents. The user can specify a file, be it text, RTF, Word etc and then have it output those contents into a PDF version which can be read by Adobe Acrobat or other PDF readers.

Tips: PDF’s have a very specific format developed by Adobe in the early part of the 1990’s. While the format is now open, it can be a little daunting to beginner developers. Your first place to start is by looking at the syntax that makes up a PDF document. You can find such reference material at the Adobe PDF Reference page. Pay particular attention to the link that reads “Part 1: PDF Version 1.7, First Edition” which outlines most of what you need to know. They have other links on this page for extensions to the standard. Once you have read the document, try to create a “Hello World” PDF to get started.

Added Difficulty: Support the extended standards or try to include an image.

Title: Quick Launcher
Difficulty: 4
Description: Make a program that allows the user to replicate the functionality of the Windows Quick Launch toolbar. The user can add multiple programs to it via an add button/menu (or drag a program’s icon to the bar) and it will add the icon there. Clicking the add button will ask the user for details about the program to add including a name, icon picture and the path to the executable file to run. When the user clicks one of the icons on this new toolbar, it will launch the program. This toolbar should float around the screen or, if the user wishes to, dock it to any side of their desktop or web page.

Tips: First create a standard form/page. Drop a button on this form/page and for the add button you will need to show another form, dialog or web page (depending on your language of

choice). Ask the user to specify the details and once they have, add the icon and setup a click event for it. To support the drag and drop functionality this form/page will need to be listening for the mouse attempting to drop items onto it. When the drop is detected, the program needs to read the information about the element dropped onto the form. This item is most likely a shortcut or an element from the form/page. The main goal here is to get access to what is being added or dropped and setup the element on the bar to then use that information to execute the program when clicked. For information on .NET drag and drop check out the MSDN Drag and Drop Overview page.

Added Difficulty: Besides just the path to the executable, allow the user to also specify additional parameters. That way not only can they open notepad when they hit notepad icon, but can have it run notepad and open a file at the same time.

Title: Quiz Maker
Difficulty: 3
Description: What program has multiple choice, random questions and user scoring? A quiz maker program! Design a program which asks the user for a list of questions, multiple choice answers for the question and the ability to specify which answer(s) are right. The program will then add those questions and answers to a file which it will read to display questions. The program will read the file, randomly pick a question and load up its questions for the user to choose from. When the user chooses an item and clicks the “final answer” button it will determine if the answer chosen is correct. It should also keep score for the end.

Tips: First we need to create our repository of questions and answers. A simple way to do this is to have your program open up and ask the user to enter questions and their answers. Each question and the answers they add will be written to a text file (or database if you prefer). Write each question and its answers on a single line (one per line). An example might be “What color is the sky? | red | blue* | green”. Your program will then read a line and break it apart to show the question “What color is the sky?” and list “red, blue, green” as the answers. The answer “Blue” is marked with an symbol to say this is the answer (which you obviously don’t show the user taking the quiz). Store that answer hidden from the user and when they select the answer compare it to the one stored. Each successful answer can be added to a counter variable. At the end of the quiz their score is this counter variable over the number of questions asked.

Added Difficulty: Support multiple answers as correct.

Title: Registry Cleaner
Difficulty: 7
Description: Computers slow down for various reasons. One reason might be because it has a dirty registry. Create a program that will go through the registry, check if any registry keys exist for a program that is no longer installed, or present on the system, and delete it. Be sure to keep a copy of the registry before making edits so that the user can easily revert to a previous version in case the registry is changed in a way that the user had not intended.

Tips: The registry is a database of program configuration settings. Sometimes when a program is installed it adds elements to this registry. When they are removed, it is suppose to remove its entries. But if the user is in the habit of just deleting files rather than uninstalling, or the uninstaller fails to clean up, it leaves old keys in the registry. Loop through the registry keys and check if the corresponding program is still there. You will want to focus on two main root keys, HKEY_LOCAL_MACHINE\Software and HKEY_CLASSES_ROOT. The local machine software key is usually modified by applications and will have application data that you can look up (like paths to the executable etc). Verifying if these paths exist and executables are present will then tell you if the program is still there and if the key needs to be removed.

Added Difficulty: Have it quarantine entries and present the user with a report of what your program found. Then provide the user with a choice to remove them permanently one by one, all of them, or none of them.

Title: RPG Character Stat Creator
Difficulty: 2
Description: Write a simple program that creates a profile for a Role-Playing Game (RPG) character. The program should ask the user to enter a class (Warrior, Mage, Rogue etc), level and Race. It would randomly assign the appropriate strength, dexterity, charisma, magic experience, stealth etc based on those attributes entered by the user. The user can hit the generate button multiple times until they get a set of stats that they are happy with. Provide a way for the user to make little tweaks to any of the stats. Also provide a print button so that once the character stats are set they can print it out and either give it to the person the character is for or save it for game records.

Tips: This program has a lot of formulas in it. There are no set formulas that you have to use and will vary based on each game. Ideally a warrior would have more strength than a mage while a mage would always have more magic points than a warrior. Think about starting out with a “base” value and a set number of “add on” points. Randomly divide up the add on points and spread them across the stats. For instance, if they have 50 add on points, randomly chosen strength might yield 23. This leaves 27 points left to be spread across the others. From those 27 if the magic points are randomly chosen as being 12, this means there are only 15 left for dexterity. You don’t want your randomly chosen values to add up to more or less than the set number of “add on” points. Look into the modulus operator to help you with determining how many points are left.

Added Difficulty: Allow the user to specify hair color, eye color, build type, starting weapons etc using drop down menus.

Title: Song mixer / DJ
Difficulty: 8

Description: Want to make your own music? Here is your chance! Write a program that will allow the user to specify various sound files that can be mixed together and recorded for playback later. The user should be able to pick various files and lay them on top of one another to create a unique sound. Save the files to wav or mp3 format.

Tips: To mix music try first by taking two waves and reading them in. A wav file will be made up of a series of signal samples that can be added together with a weighted average. Those two signals averaged together and then written as a new wav file stream will give you one way of mixing the two. If you want to do this with Mp3s, try first converting them to wav and then work on those files.

Added Difficulty: Add in cutting and joining of sound clips and also compressing the final wav file into an Mp3 so you can listen to the mix on your favorite Mp3 player.

Title: Sort File Records Utility
Difficulty: 4
Description: Reads a file of records, sorts them, and then writes them back to the file. The user can choose various sort styles and sort based on a particular field. It should ask the user to specify the file name and assumes each record is on its own line.

Tips: This is a file reading program that can start by reading all the lines into a structure like an array or some sort of List object of your choosing. Then, depending on the sort options, will sort the list based on two values in the object. My suggestion would be to use a separate comparison function that takes two values and compares them for equality. The function would then return a value such as -1 for less than, 0 for equal and 1 for greater than. Based on this returned value the sorting algorithm would then be able to determine equality values between two items and move them in the list accordingly. This can be implemented as a callback function parameter in many languages (like a comparator in Java). If the files are expected to be large, you may want to go with an algorithm like quick sort or if they are small then perhaps a bubble sort could be quickly hashed out.

Added Difficulty: Try implementing different sorting algorithms until you find one that maximizes speed and efficiency.

Title: Versioning Manager
Difficulty: 5
Description: Create a program which allows a user to keep copies of previous versions of a file. In other words, a version control system. Ask the user which files they want to be managed. Any time one of those files are edited, a copy is made and stored away in a separate location. The file is attached to a sequential version number that changes with each copy and is always unique for a given file. The user should be able to easily revert to a previous version by looking at a list of versioned copies and clicking a button called “Revert”. It will then delete the recently edited version of the file and replace it with the older one that the user specified.

Tips: This program will require that you keep a list of all previous versions of a given file. So for each file that is managed, setup a dynamic list structure that can hold a copy. When an edit is made, copy the file to another folder location (giving it a new name with the version number attached) and store that file name in the list structure. To start off with, don't try to store the actual copy of the file's contents in the list structure. Just the path to the file. When a revert is done, delete the modern file, copy that file from the stored folder to the location of the recently deleted file and give it a new version number. To help monitor the file for changes, try using an object or service like the .NET FileSystemWatcher class. For other languages that don't support a watcher service, you can periodically poll the file to see if it has changed.

Added Difficulty: Allow the user to merge changes together between two files. For this also build in a difference tool so that the user can see, side by side, what is different between the two files. Try to also think about trimming really old versions so the user never runs into an instance where they have 1000 versions when really only the last 10 are relevant.

Title: Web Document Viewer
Difficulty: 4
Description: Create a web application for a user to view a listing of all files in a given directory through their web browser. This is a Windows Explorer for the web and will allow the user to navigate up and down through the file hierarchy and carry out basic file management tasks such as creating, editing and deleting files. Provide various icons for each of the files. Make sure the icons shown for a file conform to the file's permissions (don't provide an edit icon if the file is not to be edited by that user). Clicking a file name should allow the user to get a quick view of the file's contents (for text based formats, not executables).

Tips: Start by having a quick glance at your language of choice's file/IO classes and locate classes that you would need to quickly list the contents of a folder, how you would determine between a file and a folder and finding the properties of a file/folder. Directory structure can be built using a recursive method where you list the contents of a folder, loop through it, determine if an object is a folder, and if so, call the folder listing function again providing that folder name. The recursion ends each time it encounters a folder that has no sub folders or a certain depth is reached (if you choose to build in depth control). Each time the user wants to read, edit, delete or otherwise manipulate a file, be sure to check the file's permissions using the IO functions. If permission is granted, then let the action proceed. Otherwise deny the action.

Added Difficulty: Try building in some nice client side nuances like scrolling open folders or making AJAX calls. jQuery would be a nice thing to use in such a situation.

Databases

Title: Address Book

Difficulty: 3
Description: Lost a phone number? Can't remember their address? Solve this problem by creating a program which keeps track of contacts and their information in an address book. The user should be able to create, edit or delete contacts. Creating a contact should allow the user to enter in a name, address, phone, email, notes and any extra information a user might want to keep track of for an individual contact and store it in a database. Provide a mechanism for the user to quickly look up a contact by keyword search or by looking them up using a dictionary type approach (clicking "A" to find "Adam"). Search by contact name or by phone number.

Tips: Try to start this project by determining what exactly you are going to record about a contact and the types of fields you will need to accurately capture that information. Does it make sense to have a drop down for an address? Probably not. Once you have determined all the data you will need, create a class to represent a contact. The address book itself is going to be a manager of these contact classes. The search will then be able to look through the list and access each class instance and compare it against the search keywords. Look to making a List structure that holds Contact class instances. Also determine how you are going to save these contacts between executions of the program. A text file may get you only so far, so perhaps a database will be needed. It can also help with searching. Query the database, pull out the contacts information, build a contact class with each record and store it into a manageable data structure in the program.

Added difficulty: Allow the user to specify a picture of each contact. If using a database, you may want to consider NOT storing the picture directly in the database but instead refer to the file by its path and store that. The picture then will reside on the file system.

Title: Baseball / Other Card Collector
Difficulty: 4
Description: Card collecting is a great pastime, but it would be easier if the user could quickly search their collection to see which cards they have and how many duplicates they may have of it. Create a program that allows a collector to catalog their collection. Since most collectors have thousands of cards, creating a database will be essential and being able to enter a card's information quickly will be even more important. Not knowing the type of collection they want to create, allow the user to specify the data they want to enter for each card and the field type. For instance, a hockey card may have a stat for goals while a baseball card would not have such data. Allow the user to also search the collection by card type, name or by a particular field they have chosen.

Tips: Since we don't know what the user will enter for data on a card, we can only make some basic assumptions like it having some kind of number id, name and manufacturer name (Upper Deck, Topps etc). Create a base class called "Card" which keeps the basic information of a card. For each card type, inherit from this class. The types of fields they enter, and want to keep track of, would be member data that might be best handled as a dictionary structure containing a field name as the key and the value as the value they entered. Thus a baseball card may have

a key called “Home Runs” and its value “20”. Normally these would be properties of a baseball card class, but since we don’t know what they would enter for fields, a dictionary would be a quick easy way to implement it. Make sure that each card is listed in a database table of at least 3rd normal form.

Added Difficulty: Add the ability to add a picture of the card (front and back) and create groups within the collection (1994 rookies).

Title: Budget Tracker
Difficulty: 4
Description: Develop an application where the user can quickly enter their income, expenses and savings goals and the program would calculate their budget. It would show what they need to make each month, where they can save more and calculate how long it would take for them to reach their savings goals. Store this information within a database and make sure not to save any data that could be calculated “on-the-fly”.

Tips: Think of this like a checkbook with a savings component to it. Each time a person receives income, they can create a record in the database. Thus they may need to enter data like the name on the check (or who they received cash from), what it was for, how much it was and the date it was received. Expenses might be logged in a similar manner with whom the check/cash was given to, for what, how much and when. Using the date field you can then query the database to add up records for a given month to find totals and use the name field to determine how much was given to a certain person/company. The savings portion should allow the user to specify their saving goals (how much to set aside each month and overall total they would like at the end of the Nth year) and their savings plan’s interest rate. It should be able to calculate (using an interest formula) where their money will be at any point in time.

Added Difficulty: Create at least one chart to show their savings over a specified number of years. This can help the user forecast if they are going to meet their goal or not.

Title: Database Backup Script Maker
Difficulty: 6
Description: Write an application that allows the user to create scripts used for backing up specific databases. Ask the user for the IP/host address of the database, the user name and password to access it, the tables/entities they wish to back up and any additional options (like to dropping database tables for creating, overwriting etc). Once this information is specified, the program then will connect to the desired database and create an SQL script that will recreate all the tables/entities and its data as a backup. This .sql file then will be able to be restored as a new database or to overwrite an existing one.

Tips: .sql files are nothing more than merely text files containing a series of SQL queries to rebuild tables and data. The program will need to be able to connect to the database, loop through databases which loop through tables and read its row/column data. Along the way it will

generate CREATE TABLE queries to create tables, use INSERT queries to insert data for these new tables or possibly create DROP TABLE queries to drop tables. Be sure, if the user specifies it, to capture other entities such as constraints, keys, views and users.

Added Difficulty: Give the user the option to break the file up into one which creates the structure and one that inserts the data and one that inserts users/roles. This would be advantageous for very large databases that may benefit from a staged approach.

Title: Database Translation (MySQL <-> SQL Server)
Difficulty: 7
Description: Database administrators sometimes need to change database vendors which means backing up data on one and importing it to the other. Even though most databases know SQL they may not have the same syntax. Does MySQL have an Interval data type like found in Oracle? These subtle nuances can be time consuming to translate. Develop an application which can read one database type and translate it into statements understandable by another database type. To start with something simpler, try going back and forth between MySQL and SQL Server. The result of this program is a .sql file that contains SQL statements that the target database will understand and recreates the data taken from the source database.

Tips: This is similar to the Database Backup Script Maker project but goes between databases which may have subtle syntax differences. Again connect to the source database, read the tables and rows, find out if any of the columns contain an unknown data type and make the appropriate conversion to a data type the target database will understand. Perhaps an Interval data type will need to be a float data type or maybe it needs to be split the interval into two fields. Perhaps SQL Server has views that may need to be recreated as a separate table in MySQL. Make a decision based on the available data and the types of databases involved. The user should be asked what to do for various situations in case they are encountered.

Added Difficulty: Support more database types. Keep in mind that for each one you add, the conversion choices are going to grow exponentially.

Title: Entity Relationship Diagram (ERD) Creator
Difficulty: 8
Description: Design a program that allows database admins to quickly create and visualize entity relationship diagrams (ERDs) for a given database. Allow the user to add, edit and delete entity tables. They should also be able to add/edit/delete fields and relationships and drag and drop around each entity for quick organization. In addition, allow them to draw lines, create labels and zoom in/out to get a bigger picture of the database.

Tips: A lot of this is project is based around reading the entities within a database and drawing them as a picture. So you need to 1) Be able to connect to a specified database and 2) Use a drawing mechanism, like the GDI, to draw these discovered tables, relationships and other entities. Try starting with just tables and see if you can draw a table, its fields and label it. Then

move to relationships where each relationship can be drawn as a line between the two tables along with its cardinality. Once those are done try labels, line drawing and then eventually dragging items around. Keep in mind that dragging a table will cause the table to need to be redrawn, so perhaps a table/entity drawing function you can call may actually be something wise to invest in early.

Added Difficulty: Add color or add the ability to print out the ERD in a reasonable format.

Title: Error Message Database
Difficulty: 2
Description: Put together a database of common error messages, the files that may be involved in the error, an error ID and a resolution. Allow the user to then be able to use an API to query the database and get solutions to errors they may encounter. Keep the errors related to one specific domain at first. For instance, common Windows errors.

Tips: This project is pretty straight forward. Find resources on the Internet which lists several errors, their IDs, messages and so forth. The trick here is that you should find very well thought out and well written solutions to these errors. If you build this database to be extensive enough it will surely become a resource that will be in demand. The API should be able to query for error messages/resolutions by ID, file or error message (use a wildcard for matching). This will allow users to quickly build applications around the API that will allow them to quickly interact with your database and give end users a way to look up resolutions to problems. Make an error table which contains fields for the ID, title, cause, resolution and possibly a foreign key field that links to help files or downloads in another table (that way one file may help with multiple related problems).

Added Difficulty: Provide mechanisms in the API for adding to the database from trusted sources as well as edit existing entries.

Title: Event Scheduler and Calendar
Difficulty: 5
Description: Everyone has events or appointments they must attend at some point in their life. Why not make it easier by creating a scheduler and calendar that they can use to help organize and alert them to upcoming events? Design an event scheduler and calendar that a user can use to create an event, schedule it for a date and time (using a built in calendar control or create one yourself) and have it remind them when they have conflicting events. Store this information within a database for persistence between executions of the application.

Tips: Similar to a few other projects already mentioned, this project can be tackled by creating an “Event” object that can be associated with a day and time. This object may have an ID, a message, a date and time associated with it (these can also be fields in a table on the database). Create another class that represents the days of the week, month and year and allows the user to insert these event objects into the day for a given time. This “Day” class may

have mechanisms to manage multiple events including adding new events, removing old events etc. The idea here is that events belong to a day and a time. Keep that idea in mind when you are in your initial design.

Added Difficulty: Allow the user to schedule events for multiple days. This can be done by possibly having the event scheduled for multiple day classes?

Title: Meal / Food Journal (With Calorie Calculator)
Difficulty: 2
Description: Most exercise plans ask that the user keep a journal of what they eat to know how many calories the user is taking in. Create a program that helps the user to input the type of food they are eating, look up those foods in a database and calculate how many calories their meal is. The user could also use the application to pick meal choices or ingredients that are healthy and low in calories.

Tips: This project will require you have access to some tables of information. Essentially you will need to fill the database with multiple key / value pairs where a “Carrot” is “2” calories. Or a “Cup of White Rice” is “12 calories”. You may want to go one step further and also associate values of saturated fat, sugar or salt. However, those can be things you add after you get the calorie count mechanism working. Also think about creating a “Meal” class which may manage all the different parts of the meal together and allow things like substitutions, adding items, removing items or changing portion size. This Meal class may have an internal List structure for this. Loop through the list of meal ingredients to add up calorie counts. Perhaps each of these ingredients are also a class. So you would loop through the Meal class’ ingredient list, ask each ingredient object for its calorie count and add it to a running total. The database may structure ingredients as a lookup table where you list a column of ingredients and columns for the calories, sugar and fat. Portion size may be another table since items like “cup” or “8oz” may be proportions that are used often. Think carefully about the relationships here before coding.

Added Difficulty: Analyze meals against a set target calorie intake for the day. If the calories exceed this limit, have the program make a few recommendations on substitutions. If they listed white rice, have it recommend brown rice instead.

Title: Postal Shipping Program
Difficulty: 4
Description: Design an application that asks the user for package dimensions, the weight and its destination and it will calculate the cost to ship that package with various carriers. If the user specifies a package that is 12 inches by 3 inches by 8 inches and weighs 4 lbs, the program will look up in a database for any carriers that can handle those dimensions and weight, then find their rate quotes and show the user.

Tips: Normally this type of program would be dynamic and ask carriers directly using an API. Feel free to do that if you wish, but for this exercise use a fixed list of carriers and rates for given

package sizes and weights. It would be wise to keep carriers in one table, rate types with their cost in another and then try to fit the package size/weight to one of the set rate types. For instance, if FedEx has a flat rate that anything fitting within 13 x 8 x 12 is $12.50 you could compare the dimensions to those in the database, determine that this box will fit and return $12.50 to the user. Link the carrier table to the rate types they support and then specify a cost for that carrier. Not all carriers can ship that package for $12.50.

Added Difficulty: Provide multiple options for shipping including express, ground, overnight etc so that the user can compare not only how much it will cost but which is the cheapest for their needs.

Title: Remote SQL Tool
Difficulty: 8
Description: Create a program for a user to remotely execute SQL commands on a database. The program should ask the user for an IP address, user name and password and then create a profile for it. Once connected to the remote database, the tool should allow the user to enter any SQL query they want to within their given permissions.

Tips: First order of business is to get your program to connect to the remote SQL server. Secondly, create a basic form with a place to enter text (preferably a multi-line text box) that they can issue standard commands with. The database is going to return recordsets or boolean values (true/false) for non select type of statements. When receiving recordsets, think long and hard about caching them locally. This will allow the user to easily manipulate, scroll through or otherwise filter the set without having to ask the database to do it all. One of the biggest roadblocks you want to avoid is having to pull across more data than you need to. If you are pulling millions of records each time you scroll you are going to see performance issues.

Added Difficulty: Provide buttons that can allow the user to test the format of their query (to detect syntax errors) and to quickly generate a query template (a button to put together a quick select statement?)

Title: Report Generator
Difficulty: 6
Description: Develop a tool where the user can generate reports based on a database they are connected to. Ask the user for the tables they wish to include, construct a query, pull across the records and group/sum up the data visually based on their parameters. For instance, if they put together a query to pull across a list of names and their sex, and the user wants to report this, perhaps you would group them on their sex to give a graph that shows the ratio between men and women.

Tips: Like many of the other database exercises, first connect to the database and pull across some data as outlined by the query created by the user. Cache this data! You will need to cache it so that you can work on it locally and quickly manipulate it and data mine the information. You

may need to loop through the records, keep accumulator variables or develop algorithms that will take in multiple rows of values. Once you have the data manipulated according to the user's specifications, create one or more output functions to show this data. Decide how this data is best viewed. It might be a table, it might be a standard purchase order report or could be a graph.

Added Difficulty: Develop several charting options using the drawing functions available in your language of choice.

Title: Repository App
Difficulty: 8
Description: Design an application that will be used by the user as a repository of files. These files should be versioned, stored in a database and easily searched/retrievable. During the storing process, try to find out all the information you can about the data and store it along with the data to help other users know more about a file or chunk of information.

Tips: There are several repository groups out there like Git or Codeplex. Use these as examples. Think about what it means to store a document and what kind of information might be useful to know about that document. Perhaps the file name, its size, file type and permissions. Before venturing into permissions however, try to first just storing documents and getting versioning to work properly. Once you have that part in place, permissions should be a bit easier to implement. Permissions may involve you creating user accounts and then determining which role they are in. Only certain user roles can view/download certain documents. Associate a user to a document using a joined (junction) table if need be.

Added Difficulty: Allow users to subscribe to or watch the project. Allow them to also fork a project into their own project.

Title: SQL Query Analyzer
Difficulty: 8
Description: Make a tool that the user can use to analyze queries, determine how to optimize them and which perform better than others. Does a nested query work better than an inner join in a specific situation? This tool will need to access benchmarking information to be useful.

Tips: Start with some basic queries with data that is of a good size. It is suggested to start with something like a few thousand records and a couple queries you know are absolutely correct but different in format. Benchmarking is going to be tough to get accurate but try experimenting with multimedia timers or timers with high accuracy. A timer control in VB may not be accurate enough to measure millisecond differences with great precision. MySQL comes with the BENCHMARK() function that may be of use. It executes code a given number of times and reports the time. Other databases may have similar functions you can use during optimization of queries.

Added Difficulty: Graph the results of your benchmarking.

Title: Survey System
Difficulty: 4
Description: How would you rate your experience as a programmer? Let others tell you by using a survey system. The user can enter in a question and then multiple choices which are tallied when others select them. This system should allow the user to enter multiple questions, put them together in an entity known as a “survey” and keep records of applications responses. It should then report the results as a summary at any time the creator of the survey requests it.

Tips: A couple of classes/tables can be located in the description above. One is the idea of a survey. This type of object would have an ID, perhaps relationships with multiple questions, an email address where the survey responses should be sent to and more info about the survey itself. Another entity would be the question. Each question would have a relationship to its choices. Yet another entity may be the idea of a “response” which would be the choices made by a person answering the survey questions. A table in the database could be dedicated to the choices selected and associated with the survey/questions the choices are in response to. Think carefully about the relationships here for it is easy to get a little lost.

Added Difficulty: Added in survey timing. That is, the person filling out the survey has a limited time to do so once they start the survey. Also prevent double submissions.

Title: TV Show Tracker
Difficulty: 5
Description: Always missing your favorite TV shows? Don’t have a PVR handy or its filled up with Dr. Who episodes that you don’t want to get rid of just yet? Create a program that loads up the local TV programming schedule into a database along with the times and channels it appears on. Then allow the user to enter in shows they want to be notified of when they are about to be shown. The program should ask the user to enter a list of shows they want to watch and the channels they want to watch them on (or any channel if they prefer). The program will fetch scheduling for the day and as time ticks by, it checks the database for any shows coming up that match the user’s list. If a show matches, provide a notice box to the user to let them know which show is about to start and which channel. Account for multiple shows appearing on multiple channels at the same time.

Tips: This is a multi-part project. First you need to locate a TV schedule that you can pull data from. This may be through an API or most likely a page scrape. Second part is to put that data into a database in such a way that it builds records. Channels are pretty stagnant so consider a lookup table for these. Third part would be to create a class that can handle show details like the name of the show and the message to show the user when the show is about to come on. You could create another class to manage a list of these show objects. At a specific interval, get a list of the shows that are about to come on, loop through the list of the user’s shows to watch

and see if there are any matches. Keep a list of the matches and when done comparing, show the message to the user by looping through the matches.

Added Difficulty: For the show class, try also setting a description of the show (obtained from the TV schedule) and perhaps let the user enter channels to watch for the show on.

Title: Travel Planner System
Difficulty: 6
Description: If the user is wanting to plan a trip to a foreign country, or even a nearby state/region, they could use a system to gather all the data with and plan the trip! Develop an application that allows the user to enter a destination and pull across several bits of information from different sources. Perhaps the travel planner loads a map of the area while also showing which hotels they can book at and the price. Provide information on transit, culture, airfare and even down to the smallest details like the local restaurants and their cost. Organize all this information into a database. Either pull the info from various websites and store it in the database, or manually enter this information into the database for several locations.

Tips: Obviously pulling information for airfare and hotel will be best done automatically. However, the information regarding culture, weather, currency or whatever else can be manually entered too since it changes very little. Perhaps you could even provide the database and build an API on top of it for other users! But that is another project. Keep all data in the database based around the idea of a location. This location entity will also be a perfect place to start building an object. Perhaps you can give the location a unique ID to help. Before coding, identify sources for all the information you want to pull from. Then look for an API or way to quickly extract information. Keep in mind that some of these sources may ask you to pay, but there are free sources out there as well if you look hard enough.

Added Difficulty: Allow the user to be able to book certain services through the program. Maybe they can buy their airfare or hotel through the application.

Title: Web Board (Forum)
Difficulty: 7
Description: Design your very own web forum. The application should have various forum categories, with the option of sub forums, users and their roles. In addition to this, it should keep track of topics and replies and track things like posts per page and a paging mechanism to move through popular topics. This is an open project where you could add many advanced things. But start small and stick to the basics for now. Allow the user to create an account, create/edit/delete forum categories and posts, add replies and page through posts. All posts and content will be kept track of through a database.

Tips: Before you start with this, thinking heavily about relationships between the different elements. A web board contains many forums, each forum contains many threads, each thread contains posts. Users can be admins, moderators, regulars or possibly banned. When designing

the relationships between each of these entities, always keep in mind who is going to be able to access and see various items. Admins can edit all posts, but perhaps moderators are associated with only a particular forum thus trying to edit posts outside of that forum is not allowed. Look to other forums already created to see how they do it and associate these objects together.

Added Difficulty: Allow users to insert pictures, show code snippets, add emoticons, allow the admin the have an interface for managing banned accounts, build in a full featured editor... the list goes on.

Graphics and Multimedia

Title: Bulk Picture Manipulator
Difficulty: 6
Description: Design a program that asks the user to select several picture files from one or more folders and goes through each one applying an image manipulation. These manipulations could be to turn each picture to a grayscale version, re-size them to specified dimensions, watermark them or perhaps apply some sort of filter.

Tips: As with any process that may take some time, it is advised that you first start by focusing on one image manipulation and when that is working, put it into its own thread and replicate the procedure. We want to keep these manipulations on their own threads so that it will process in the background and not bog down our user interface. Look up the image manipulation functions for your language of choice, then how you would go about spawning a thread. This may not be an ideal tool to work on for web languages like PHP unless you can actually have the PHP pass the images to an extension written in another language.

Added Difficulty: Support more manipulations like blurring/sharpening, cropping or altering colors.

Title: CD/DVD Burning App
Difficulty: 7
Description: Make an application that allows the user to burn their own data or music CDs. If they choose to make a data CD, ask the user which files they would like to burn onto the CD and then burn those files using their CD/DVD burner. If they choose music, allow them to select various mp3/wav files to burn to the CD. Keep in mind that many older CD players may require that the files be burned in raw audio format.

Tips: So there are two things we need to do here, burn data CDs/DVDs and burn music CDs that can play in most CD players. If you are using a .NET language it is suggested that you take a look at IMAPIv2.0 as an API that burns images to disc. This API can also do things like burn ISO images as well. Other languages may have to find the format needed to burn data and do it the hard way or find a library that can burn according to that format. Music CDs, that most

players can understand, is in a format called CDA or Compact Disc Audio. This is the format mostly used by commercial CD makers. Create a buffer structure for the data, read the data files, fill the buffer and then write the buffer to the CD/DVD when ready. Make sure you get the structure right first before attempting to write to the discs otherwise you will be throwing a lot of them away.

Added Difficulty: Try increasing the speed of writing or supporting various formats that the user can then choose from.

Title: Image Browser
Difficulty: 4
Description: A user may want to browse their collection of images. Help them out by creating a small program that can be used to browse large collections of images on their computer. The program should be similar to the Windows Image Viewer app that comes with Windows, but more robust and can support various image formats. The app should take into account the size of the image and scale it accordingly to fit on the user's screen. Provide a way for the user to quickly click through from one image to the next.

Tips: This project focuses around image reading and display. Based on the type of file, it may require more or less work to show. If you are using .NET, try playing around with the Image class (System.Drawing) and load the image so that it can be displayed in something like a picturebox. Perhaps you might want to use the GDI to draw images onto a form. Java users can take a look at similar classes and using the paintComponent() method to paint onto a JPanel. PHP users will want to take a look at the GD and image functions section.

Added Difficulty: Provide a smooth transition from one image to the next. Perhaps a fade out, fade in transition?

Title: Import Picture and Save as Grey scale
Difficulty: 5
Description: Create a program which will ask the user to select an image and, after it is loaded, will remove all the color turning it to a grayscale image (black, white and greys). The program should then allow the user to save the image back to disk.

Tips: Start by isolating each pixel of the image and the individual RGB values that make up its color. Then by applying a formula such as taking 30% of the red value, 59% of the green and 11% of the blue values and adding them together you can achieve a grey color which is the same luminance of the original colored pixel. Once you have this color, it is just a matter of replacing the color pixel with the new one. Look at using the Bitmap.SetPixel in .NET as an option for this.

Added Difficulty: Since the program now turns the image grey, can you change the grey image into some other kind of color scheme? A special "colorized" feature?

Title: MP3 to Wav Converter
Difficulty: 4
Description: Design an application which asks the user to specify a Mp3 file to convert to a Wav file. The user should be presented with a dialog box that asks them to specify the file. After selecting it and clicking OK, the program will then progress through the file converting it to Wav. At the end of the process, a new Wav file is created which is a replica of the original Mp3.

Tips: One of the easy ways to do this is through the use of a library. One such library for C# is the Alvas Audio library which you can use to do this. You can even view a tutorial video to see how it is done. Other languages may have similar libraries. Keep in mind that Mp3 is a bit like a compressed version of Wav. During the conversion you will see the result Wav will be much larger than the original (about 4 times larger). So if you have a 5 MB Mp3, plan for it to be close to 20 MB at the end. You may also want to take a look at using threads for this just to keep the process off the main GUI thread and let it convert in the background.

Added Difficulty: Now that you can go from Mp3 to Wav, have it convert the other way.

Title: Mind Mapper
Difficulty: 5
Description: Write an application where the user can create mind maps. These are quick diagrams that link together words or ideas around a central theme. The idea is to create a map that quickly explores a theme to gather thoughts and focus the mind. The program should allow the user to draw “bubbles” and lines which link together related ideas. Also provide a mechanism for dragging around the ideas and form a structure of interrelated ideas.

Tips: Since there is drawing in this project, first explore the drawing tools available in your given language. Each idea “bubble” could be its own control or panel which you can draw onto and could provide an object to attach to for drag and drop functionality. If you create a “Diagram” class that keeps track of all the bubble objects in it, this would make it easier to loop through and apply effects.

Added Difficulty: Allow the user to quickly paste in pictures or break out one part of the map into a new map of its own.

Title: Mp3 Player
Difficulty: 6
Description: Make a program which can play a chosen mp3 file. The user selects “Open file” and chooses a file. The program will then read it, display the file details and then play the file. Also add a button that allows the user to create their very own playlists. They can add one or more songs, save the list and the player will play each song through the list. When the playlist is saved, it creates a playlist file. This file can be loaded into the player at any time.

Tips: Mp3 file information can be obtained from ID3 tags. Read these tags to pull out the information such as artist, track number, album or even album cover art. Info such as bit rate and frequency can be read from individual frames in the file. Playing the actual file will require you to read the file as a bit stream and pull out the frequencies and amplitudes from the compressed audio data. Depending on the language you choose, this may require you to find a library dedicated to this purpose. For the playlist functionality, look to file writing and reading functions. The playlist should be written into the file as a certain structure (maybe XML?) which can then be easily read back in.

Added Difficulty: Try to support other formats if you can. How about .au files?

Title: Screen Capture Program
Difficulty: 5
Description: Make a tool that the user can use to capture the current screen and save it as a picture. The tool should sit in the system tray and when the user right clicks the icon it provides a context menu which has the option to capture the screen. It should then pop up a dialog box to ask the user where they would like to save the screen capture on the user's hard drive.

Tips: To help facilitate this, you may need help from the operating system. For some help look to the file user32.dll which has some window handle functions that will help snag the contents of a given window. Some of the functions to look at include GetDesktopWindow() and GetWindowDC() among others. You will need to get a handle to the window you wish to copy, obtain a context handle for it and with it be able to copy the window contents into a new graphic file for saving.

Added Difficulty: Provide the options to copy a portion of the screen, a specific window only and copy to clipboard instead of to a file.

Title: Screen Saver
Difficulty: 4
Description: Write a screensaver that the user can run while they are away from their computer. Make sure that this screen saver can be chosen as a possible screen saver through the OS and when run it should cover the entire screen.

Tips: Your typical screen saver is a program that runs while the user is inactive. They were designed in the early days when monitors use to be cathode ray tubes (CRTs). One image sitting on the screen for long periods of time would eventually "burn" the image into the monitor and ruin it. You can make yours as a program which takes up the entire screen. Look to the System.Windows.Forms.Screen class in .NET to help with getting the screen properties. After your program takes up the entire screen, design an event/listener that will listen to the keyboard/mouse for any action. If an action is present, you can shut off the program or hide it in the tray. The program itself could show images as a slideshow or create an animation.

Added Difficulty: Make the screensaver show several images that bounces around the screen.

Title: Signature Maker
Difficulty: 1
Description: Make a program which gives the user the ability to generate either an email signature or forum signature for their posts. Ask the user to enter in some free form text or HTML and control the placement or its alignment. In turn, the program will generate text that they can copy and paste into their email program or into their profiles on a forum.

Tips: This can be a very simple program. Provide a text box and a few buttons on a form. Allow the user to enter text into the textbox as they see fit. Make sure you keep track of the lines and carriage returns (those may need to become HTML
 tags for web forums). When they click a button, add on some more text to what they have written (in other words concatenate). The goal here is to allow the user to build up a paragraph of text/HTML which is formatted in such a way that they can then just copy and paste it into the program of their choice.

Added Difficulty: Allow the user to insert graphics such as a userbar, banner or icon. Remember that the user may need to upload the graphic to a web forum if they wish to use it there.

Title: Slideshow
Difficulty: 4
Description: Create an application which asks the user for a series of images and presents them as a slide show. The user can click through each slide as quickly as they want with Next / Previous buttons, include captions, the ability to maximize to the full height/width of the screen and control the timing of how long each picture is shown.

Tips: First of all we need to ask the user for pictures so we will need to present some sort of dialog where they can do that. In .NET this would be your classic open file dialog control, in Java this would be an open file dialog form as well. Other languages may need to create their own dialog for doing this. Once we have the collection of image files, we need to put them in a structure that will make accessing one in turn easy to do. This could be an imageList control or a classic List data structure will do. The next and previous buttons will control an index variable used to pick which image from the list is shown. “Next” would increment the index variable while “Previous” would decrement it. When you reach the end of the list, you need to roll the index back to the beginning. Same with the first slide needing to go to the end if they hit “Previous”. The timing feature will require a timer control which will essentially execute the “Next” button functionality at a given interval.

Added Difficulty: Build in transitional effects such as a fade in or fade out or some kind of wipe effect.

Title: Stream Video From Online
Difficulty: 7

Description: Design a program that will prompt the user to enter in a URL to a video file online and then stream it through the program. This URL could point to a YouTube video or it could be some sort of Flash/Avi/Mpeg video that they have hosted on a remote server. For simplicity's sake, start with the YouTube videos first and then move on to other Flash videos.

Tips: Once the user has identified the video stream, your program will need to be able to read the stream, buffer it and then play it using the appropriate codec or player. For .NET users you could make this much easier on yourself by using the Windows Media Player control or if you have a particular video library in mind, you could use that. Other languages typically have multimedia libraries. For example, Python has PyMedia or other libraries like Pyglet which may prove useful. Some sources like YouTube have their own API which could be used to embed the video directly into an application and fetch information about the video like title, run time or frame rate. So give those a look. Do a bit of research here before starting the code otherwise you may find yourself at many dead ends.

Added Difficulty: Provide a way for the user to also download the file while they are watching it. Just in case they think the video is good enough to keep locally on their PC to watch again later.

Title: Traffic Light Application
Difficulty: 2
Description: One of the most basic of programs, create an application that mimics a traffic light. Draw a traffic light with red, yellow and green lights along with three buttons. One for red, one for yellow and one for green. When the user presses one of the buttons, the corresponding light lights up and the others turn off. No two lights should be on at the same time.

Tips: The heart of this program is the mechanism for turning on lights and turning off the others. There are many ways you can achieve this, but one of the easiest tends to be a simple if-elseif-else structure which detects which lights need to be turned off and on. If the user presses the yellow button execute "If red is on, turn it off. Else if green is on, turn it off." Then the only part that is left is the drawing of the image. Once we have turned off the appropriate lights and turned on the right one, we need to read the state of each light and draw the appropriate colored circles. For this look up the paint/paintComponent method in Java and how to use the Graphics object in .NET/Java.

Added Difficulty: Make the lights change automatically after a certain period of time to simulate a real traffic light.

Title: Turtle Graphics
Difficulty: 4
Description: Make an application that instructs a turtle icon on the screen to draw various shapes based on user input. For instance, if the user issues the command "drop pen" the turtle will start a line. If the user then issues the command "move to 1,1" it will then draw that line from its current position to coordinates 1,1... which then makes a line. The user can then tell the turtle

to “lift pen” where it will then end the line. The program should allow the user to instruct the turtle how to draw lines, move to various coordinates and drop or lift its pen.

Tips: This type of program is great if you wish to learn how to take in user instructions, parse them and then translate them into actions the turtle on screen does. So the first part is to come up with a syntax for commands that the program can parse. For instance, perhaps the command is “DROP” and the object is “PEN” in which case you can tell the turtle to drop its pen. If the user enters “DROP MARKER” it would see “DROP” and understand it, but would not understand “MARKER” so it would issue an error. Once you have a function that can parse various commands, all that is left is to instruct the turtle what to do. It is suggested that you create various functions that you can call to control the turtle. You could even make a Turtle class and have various methods to control it.

Added Difficulty: Have the turtle draw a star from its current location with one command.

Title: Wallpaper Manager
Difficulty: 3
Description: Develop a program where the user can modify their computer’s desktop wallpaper quickly and efficiently. The program should allow them to add wallpaper images one at a time or in bulk. It would then let them flip through multiple wallpapers at a time. In essence, a wallpaper gallery where the user then picks one to have it loaded automatically. The user should also have basic wallpaper functionality such as position (like center, stretched, full size etc).

Tips: Since the desktop is controlled by the OS, it is reasonable to assume that any access to it is going to require the help of the OS’ API. First stop should be to look up which functions are needed to modify the background of the desktop. For Windows you will want to take a look at the function SystemParametersInfo() in the User DLL. With this function you can pass it parameters for setting the wallpaper. For displaying the images in a gallery look at possibly using a panel to draw the graphics on or perhaps a control like ListView Control with the large icon setting.

Added Difficulty: Provide a preview function that lets the user quickly see what it would look like prior to setting the background.

Title: Watermarking Application
Difficulty: 5
Description: Design a program that gives the user the option to load in a .bmp/.gif/.jpg or .png and then create a watermark on top of it. The watermark should be either text or some kind of logo... perhaps both. Watermarks are those faint images you see on some copyrighted images so that people can’t simply steal them without permission. Give the user the option to control where to place it, how big to size it and how faint to make it.

Tips: Watermarking is simply drawing an image on top of the image with a certain level of opacity (how transparent it is). So first we need to provide the user with the ability to select the image they want to work on. Once they have done that, we need to load it into an image class. We can then get the drawing context from the panel or control and draw on it. .NET makes this simple by using the Graphics Object's FromImage() method. Java can do this by creating a BufferedImage class and then using its createGraphics() method to get a Graphics object that we use to draw on it.

Added Difficulty: Try saving watermark presets so that a user can create a logo watermark, save it and then automatically add it to any image they want at a later time without needing to reload it. You could also use this ability to then do bulk water marking.

Title: YouTube Downloader
Difficulty: 6
Description: Similar to a previous project, this project's focus is to simply work with the YouTube API alone and embed videos right into an application. Feel free to make this a user control that other programmers could use to then embed video (and its playback controls) directly into their forms. Provide full video control as well as properties for controlling sound and loading/unloading videos.

Tips: First stop is the YouTube Data API which is what you will want to start familiarizing yourself with. If your language of choice is a web language, take a look at the player API instead which helps with embedding videos into websites. However, you can still use the data API if you want more functionality. YouTube has also provided developer guides for various languages such as Java, PHP, Python and .NET. Take a look at those to help guide you through the requirements and view some example code. Keep in mind that we want to start off with the most simple approach possible until we get things running and then build onto it. There is a lot of features that can quickly cause a new programmer to become confused and in over their head.

Added Difficulty: Provide search capabilities into your application for searching YouTube videos to load. Allow the user to then scroll through search results and pick videos that are then launched.

Games

Title: 2D RPG
Difficulty: 8
Description: Create a 2D role playing side-scrolling game much like the Legend of Zelda or the first Super Mario Brothers. The game should have a plot, a main character and multiple monsters to kill. Introduce hidden items, puzzles and treasure chests. The character should be able to move through the world from screen to screen, open chests for treasure, discover items they can use (like a better sword) and attack monsters. The character and monsters should have hit points, defense points and engage in battle with one another.

Tips: Start off with the plot. Create a storyboard of how the player is to progress through the game keeping in mind of characters, places, items and where major events may happen (like a boss fight). The better and more detailed the story, the easier coding the game will be. Second, create any formulas for assigning damage or game mechanics along with collision detection (so your character can't simply pass through a wall). This will often need to be tweaked to balance the game properly. Third, create your sprites and graphic resources. Fourth, start by creating the main game loop where all drawing will take place. If you are using a special tool like XNA or DirectX, by all means follow their instructions for proper game setup. In the game loop break out the tasks such as determining character stats, positions, chosen sprite animation and last should be drawing. To prevent a flicker, be sure to use Double Buffering.

Added Difficulty: Include magic spells which will alter a character's stats. This could be increase/decrease damage, magic points, armor rating etc.

Title: Battleship
Difficulty: 6
Description: Make the classic game of Battleship. Two players are given a grid representing the ocean. At the start of the game each player is given several ships which occupy two or more of these squares. Each player places their ships within this grid in an unknown location to the other player. They can be placed horizontally or vertically. After both players have placed their ships, the game starts with one player picking an opponent's square in the grid. This is the player bombing that area. If that square is part of a ship, the other player has to announce "hit". The other player then takes a turn to do the same. Once all squares of a particular ship have been hit, the player announces "sunk". This continues back and forth until one of the player's ships have all been sunk.

Tips: Think about the grid portion of the game for a second. A grid might be best represented as a 2D array. Each square would then correspond to a cell in that array. The top left of the grid would be board[0,0]. Each square in this grid will have one of four states... a ship not hit, a ship hit, open water or open water that has been bombed already. This state is what we will check as we loop through the board to check current status of the game. The loop to do that will be a nested loop. After each turn, if a hit was made, we need to loop through the array and see if the hit resulted in a sunk ship or if it resulted in a sunk that won them the game. The trick here is registering the hit and checking if all connected ship squares have been hit. How you do this is up to you. You could store the range of squares for each ship and keep track of the hits on each or you can build a function which checks surrounding squares of a hit.

Added Difficulty: Build in a small animation or sound when a hit happens.

Title: Black Jack
Difficulty: 4

Description: This is a card game where a player and a dealer are served one card down and one card up. The goal is to be as close to the point total 21 as possible without going over (known as a bust). Whichever player is closer wins. A player can ask the dealer for additional cards by saying “hit” in which the dealer gives them another card face up. Each numbered card is face value, face cards are worth 10 points and an Ace can be either 1 or 11 points. For instance a king and an ace would be 21. A ten and a nine would be 19. Two queens would be 20 and 2 jacks plus a 5 is 25 and would be a bust.

Tips: Card games are always great for teaching programmers about enumerations (enums). The suit of a card and its value can be stored in enums and referenced as lists. Each card dealt can be of type “Card” with a face value and its point value from these enums. Together multiple cards form a “hand”. So what we need to do is create a hand object which Card objects can be added to it. We also need a mechanism for looping through the hand, asking each card its value and summing it up into a hand total value. If this value is above 21, they bust. If it is under 21, but is better than the dealer’s hand total, they win. Otherwise they lose. Make sure that at the end of each “turn” the user is given the choice to hit or stay. Once they stay or bust, the cards are evaluated and the game ends.

Added Difficulty: Make this graphical so that the player can see the cards being dealt as pictures.

Title: Breakout
Difficulty: 8
Description: Develop the classic game Breakout where a set of blocks are put together at the top of the screen with a paddle at the bottom. A ball is thrown into play and the user uses their paddle to hit the ball towards the blocks, causing them to break or bounce off (based on the type of block). The user must keep the ball bouncing between the paddle and the blocks until either all the blocks are destroyed or the player misses the ball and the ball falls through the bottom of the screen. Provide at least three types of blocks, one solid block that won’t break, one that breaks on first hit and another that requires the ball to hit it twice to break. The player moves to the next level when all the blocks that can be broken are. They beat the game when there are no more levels. They lose when they run out of “lives”.

Tips: Gather your sprites and a standard game loop. Also, before you start, create the algorithm that will determine the trajectory of the ball when it hits a given surface (block, wall or paddle). You will also need a formula for collision detection to know when a ball has hit a surface. Consider making each block its own instance of a “block” object which can keep track of the number of hits it has received and also able to trigger its “breaking” animation when the hit counter has reached its threshold. On each pass of your game loop, you are going to read in the user’s input (left or right as they control the paddle) and then draw the blocks (if something needs to be done to them), the position of the ball and the new position of the paddle. When the ball meets a surface, use your trajectory formula to set the ball on a new course.

Added Difficulty: Create special blocks that give your paddle special abilities. Make a block that allows your paddle to become magnetic and holds onto the ball until the user lets it go. Create another that gives your paddle guns to shoot blocks while it waits for the ball to come back.

Title: Chess or Checkers
Difficulty: 6
Description: Program a game of chess or checkers. If programming chess, make sure that you have a set of black and white pieces on a standard chess board. Each piece has specific moves and can do only certain things. For instance, the queen can move in any direction as many open squares as possible (granted they are not blocked by a piece). A king can only move one square at a time. If you decide to program checkers (which is typically the easier of the two) make sure you setup two groups of pieces where one is red and the other is black. Most pieces move the same and they capture by jumping over opponent pieces diagonally... also capable of multiple jumps. Pieces which reach the other side of the board become “crowned” and can move in either direction.

Tips: Programming chess can be more complex than it may seem at first. Focus on the idea of a base object we can call “Piece” and inherit from that into each individual piece, overriding a method like “move”. Since each piece moves differently, each piece will have its own code for how and where it can move. Checkers too can have a “Piece” base object but its functionality is a little more limited since each piece pretty much can do the same things as any other piece. In either game, also think about a “Board” class which will contain, add, remove and manipulate your Piece objects. It will also have functionality that will help with validating the boundaries of the board (so the queen can’t simply slide off the edge) and will be in charge of keeping the state of the game. You may also want to build in functions that can quickly give you a “snapshot” of the current board. This will help you in debugging to know if your pieces are where you expect them. Look to have one function which loops through the board printing the location of pieces.

Added Difficulty: Provide the functionality for special moves like castling or make it visual if you decided to start out with a console application. Try building in also an online component so that you can play someone over the Internet.

Title: Connect Four
Difficulty: 5
Description: Recreate this game classic of Connect Four. Two players are given a vertical game board which contains several slots to put pieces in. Each player takes turns dropping in a token piece into the top of the board where it then falls down and fills the first available slot of a given row. The goal is to get four of one player’s pieces to line up vertically, horizontally or diagonally on the board. Think of it as an expanded vertical version of tic tac toe.

Tips: As with most board game setups like this, the best approach is to think up a data structure to represent the board. Typically the best one to come to mind is a two dimensional array. To

help manage this, your “Board” class should wrap this array up and control pieces locations and adding/clearing the board. It is the “managing” class for various connect four token “objects” (aka piece). Each token object will have a color associated with it at a bare minimum. The board class will also have functions to evaluate if a win has occurred or if a tie has happened (when all pieces on the board are full but no one has connected four in a row). Evaluating if a win has happened simple takes a loop through the board looking for four pieces sharing either a row, column or diagonally from one another. Think long about how you will check these conditions before you start coding. It will make the process much simpler to implement and avoid problems later.

Added Difficulty: 3D connect four maybe? How about adding an online component to play friends? Multiple games to create a leaderboard?

Title: Craps
Difficulty: 6
Description: Develop the gambling game of Craps. The idea behind this game is to roll two dice and establishing a “point” total or to hit 7 or 11 (known as a natural) to win while avoiding a 2 (snake eyes), 3 or 12. Once the point is established, the roller must keep rolling until they hit that point total or until they roll a 7 in which case they lose. To understand the full rules of the game, I suggest reading up on the game of Craps from Wikipedia.

Tips: This is a dice game where rolls are made and point totals calculated. This idea right here should automatically scream “random number generation”. So in your language of choice look for ways to generate random integer values (since you can’t have 3.14 on a die). Once you have a generator, store it in its own function to call on it for each die you need to roll. Meaning if you have two dice then you call the function twice. The point total is then added up and evaluated against the various point totals. If a “point” has been established, this dice total will be compared to the point. If it is a 7, and it wasn’t the coming out roll, you know they lost. Rolling dice an unknown amount of times (until they hit their point or crash and burn) should also be a hint that you will need a loop where the condition turns false... aka a while loop.

Added Difficulty: Build in a betting system so that players watching the game can place bets based on the current state of the game. Keep totals of who has won/lost.

Title: Crossword Puzzle
Difficulty: 8
Description: Make a program which displays a crossword puzzle that the player then has to fill in. They are provided clues for each row/column of the puzzle to help them find the hidden word and fill in the table as they progress. Various words will provide further hints to other words since many of the words will share letters.

Tips: Here we have another grid style game where things would be best represented by a two dimensional array. The smaller the puzzle board, the easier things should be for testing (but

don’t make it too small). Try first by creating a list of known words and their associated clues. This can be done with an array/vector/list of its own. Then have the puzzle generator choose various words from the list that can be placed in the puzzle either horizontally or vertically and that share all the needed letters and size. Some things to think about: The length of the words. Do they fit in the width of the puzzle board? One word should not immediately follow another word otherwise it will look like one long word. Some squares are going to be empty. If you put in one word horizontally, its letters will need to be used in the proper position of words going down vertically. You also don’t want to have the same word more than once per puzzle. The last tip, make an extensive list of words/hints. The more words you have to choose from, the more likely that they will fit together. Take a look at real world puzzles for inspiration and word choices.

Added Difficulty: Expand the board to make a super crossword.

Title: Find Way Out of Maze
Difficulty: 3
Description: Create an algorithm where, when presented with a maze, the user can find the quickest route through from the start to the finish. This should work for any maze with a solution and with 90 degree turns (that is they have only 4 possible directions of travel while in the maze).

Tips: When you boil this problem down you will quickly realize that mazes are merely a series of decisions. Some will lead to a dead end and some will get you closer to the finish. One solution is to think recursively where the user will try each route systematically until they finish or hit a dead end. In the case of the dead end, the algorithm will retrace its previous steps and try a different direction until all choices are exhausted where it will retrace back again. To illustrate this idea, think of a tree data structure where each branch (node) is a different direction in the maze. If you end up on a node with no children, you have to go back up to the parent and try another child node. When you run out of children on that node you then have to go back up again to its parent. For testing purposes, start with a small maze with a known solution already worked out by hand.

Added Difficulty: Provide a way to display a given maze and then draw the solution once it is found.

Title: Frogger
Difficulty: 5
Description: Recreate the arcade classic Frogger. The goal of the game is to direct a frog across rivers and highways to get to the other side without being hit by a passing car or falling into the river where it will be washed downstream. Each of the cars (and the logs/turtles used to jump across the river) are moving at different speeds and timing is key.

Tips: There are multiple ways to accomplish such a game but one thing that is key is the animation and moving various objects at different speeds. Think about how each item moves

given each repaint of the game. One truck may move 1 pixel at a time while a log may move 3 pixels in the same time frame. In Frogger the frog must miss being hit by a vehicle (so we need a collision detection algorithm) yet in the river they must be able to land on floating logs. Perhaps a few functions detecting if a vehicle sprite is in contact with the frog and another to determine if the frog is safely on a log (the two sprites touch). We must also keep in mind that the frog won't be in both spots at the same time so you can't test if they are missing a vehicle and on a log if the frog is only in the street. Draw out the game dynamics on paper first before coding otherwise you may chase down a solution that just won't work and waste all that time.

Added Difficulty: Introduce your own challenge in addition to the cars and floating logs. Perhaps has to avoid other hopping frogs?

Title: Game of Memory (aka Concentration)
Difficulty: 3
Description: Make a program that relives that childhood game of Memory. In this game a player is given X number of cards where each card matches one other. The cards are randomly distributed in a grid fashion. The player is then asked to turn two cards to try and form a match. If they match they are left face up and turned back over if the cards don't match. The player then attempts to make another match with different cards. They win when all cards are face up and all matches have been made. Make sure the number of cards in the game is always an even number.

Tips: A grid setup should scream out one obvious data structure, a 2D array. The user can choose one card from the array and then another and an algorithm should compare the two cards chosen to see if they match. If they do, the cards are removed from the array (or set inactive) so that they cannot be chosen again. So think of a card class, how you can compare to cards (perhaps a special comparator function or callback function) and how you would randomly shuffle the cards in the array. You will also need a function to determine if a win has occurred. This can be done by looping through the entire array and see if any are still active.

Added Difficulty: Increase the number of cards in the game making sure that an even number is always present. Try introducing your own rule like if a card is turned over and not a match, those cards can't be turned over again for a certain number of turns (unless it is the last pair).

Title: Guitar Hero Clone
Difficulty: 9
Description: Develop a clone of the popular console game Guitar Hero. The goal here is to be given a variety of songs where the user is to play the guitar (or could be drums) and must match the notes as the song is played. The player must hit a certain percentage of the notes (by tapping certain keyboard keys in tandem with the note being shown on screen) to advance to a more difficult song.

Tips: This can be a difficult game to get right because timing is everything. You must be able to detect when a note is shown, how long it is shown, if the player hit the right key at the right time and if the note is held that the player holds it long enough to consider the note played successful. The score will be based on the number of notes played in the song and how many of the notes the player has correctly identified. So one suggestion would be to find a simple song (hot cross buns?) and setup all the notes to be played and displayed on screen. Then, identify which keys the game will use for each note. Match the key to the note by showing the note, playing the sound and then waiting X number of milliseconds/seconds for the proper key to be pressed. Most languages can detect key presses using a key press or key down event. Those who are developing on the web may need to use JavaScript to get this type of interaction. They may also want to implement something with the mouse.

Added Difficulty: In the background show various animations of the band playing just like in the real game. Also add various songs of your choice and set them up with varying levels of difficulty.

Title: Hangman
Difficulty: 2
Description: Hangman is a game where a word is secretly chosen and the player guesses letters to fill in the word. Each correct guess fills in that letter in the word. Each wrong guess results in a stick man hanging from a noose to be drawn. Guess too many wrong letters and the full picture is drawn and the player loses.

Tips: It is all a matter of taking the letter the player guesses and looping through the word comparing it to each letter in the word character by character. If the letters match, you show that letter to the player. If you reach the end of the word and no letters have been matched, it is a wrong guess and you kick off drawing the next part of the picture. It is suggested that you create a function which takes in a letter and a word as parameters and returns true/false if the letter was found. In this function you could have a loop which takes the letter, loops through the word and matches it to each letter. Remember that strings are often treated as an array of characters. Most languages have a function to fetch a single letter from a string. Another function you will need is one which can draw the appropriate shapes for the hangman. Keep track of how many wrong guesses the player has done and use this number to determine which part of the picture to draw.

Added Difficulty: Increase the length of the words and choose more complex unknown words to have the player guess.

Title: High/Low Number Guessing
Difficulty: 1
Description: This game involves a secret number being chosen by the computer within a specific range (let’s say 1-100) and the player then tries to guess this number. If the number

they guess is too low, the program will say “too low” and if it is too high it will reply with “too high”. The player wins when they guess the correct number.

Tips: First thing to look at is generating a random number. Despite the language you choose, most of the time a random number can be created using a random generator function or object. .NET has the “Random” object and C++ has rand(). Use these in a stand alone function to generate a number in the given range. Make this a generic enough function so you can save it for use with later projects. Once you have a random number chosen, ask the player for a guess. Compare their guess to this random number chosen. An easy if statement can be used to see if it is too high or too low. If they are equal, tell the player they win and restart the game.

Added Difficulty: Put a limit on how many wrong guesses they can have. Too many and the game ends with “You lose”.

Title: Ice and Dice
Difficulty: 2
Description: Have the user roll three dice. Each face that has a dot in the middle is said to have a hole in the ice. Each dot not in the center is said to be a polar bear. Have the user enter the number of holes and polar bears in a given time period. So for example if you have the side with 1 dot (in the middle) it is 1 hole and no polar bears. The number 2 has no dot in the middle but two dots in the corners so it has no holes and 2 polar bears. The number 3 has one dot in the middle and two dots in the corners so it has 1 hole and 2 polar bears. The number 4 has no holes and 4 polar bears. Five has 1 hole and 4 polar bears etc.

Tips: We need to use a random number generator to come up with the values for each given die. So when the roll is done, we generate three numbers between 1 and 6. With these numbers we can figure out which value has a hole and which has polar bears (or both). For figuring this out, it is suggested that you create a function which takes an integer and compares it to 2, 3, 4, 5, 6 to know how many polar bears it has. Then if that integer is 1, 3, or 5 we also know it has 1 hole. So for instance if a die is rolled and is a 5 we know that it has 4 bears + 1 hole. Keep these values and start a timer (timer object, sleep() etc) and if the player puts in an answer, match it up to what is stored.

Added Difficulty: Put in some kind of animation showing the bears and the hole in the ice.

Title: Magic 8-Ball
Difficulty: 2
Description: Create a game that is like shaking and reading a Magic 8-ball toy. The program should have 8 or more vague yes/no/maybe style answers. The user can ask a question, press the “shake” button and it will randomly return one of those answers as if it was answering from the cosmos!

Tips: Store your answers in a simple list structure (like an array, vector or other type of list) and simply use a random generator function to pick a number. That number will be an index value for which you use to fetch the value out of the list and return it to the user. This number should be an integer value. So for instance, if you have 8 answers in an array and the user presses "shake" it will randomly choose a number between 0 and 7 (remember arrays are zero based in most languages) and uses that number to access the appropriate answer string from the list.

Added Difficulty: Create an animation that shows the "floating" answer appear in a picture of the magic 8-ball.

Title: Monopoly
Difficulty: 7
Description: Recreate the old fashion board game Monopoly! The game should allow up to four players where one must be a real player and the rest can be computer players. The goal is to develop multiple properties, earn cash, bankrupt your foes and bargain your way to the top! Players move tokens around the edge of a square board which contains multiple squares representing various properties, utilities (such as railroads, waterworks etc) and random chance squares. Players landing on an non-purchased property can then buy that property. Landing on the property of another player will result in paying rent depending on the state of that property. Learn more from the article on Wikipedia.

Tips: This game needs to be pretty graphical to make it fun. So start by drawing the various properties and assets around the edge of the board. Each square can also be represented by a class that lists various details such as its price, whether or not it has buildings, is it in foreclosure, its color, is it purchased or not and if a player is currently sitting on the property. You will also want another class that represents a monopoly on a given property. This class would represent two or more properties in the same color and if they are all in good standing (no foreclosures). Another class would be needed to represent the chance cards and possibly another to represent player pieces. Fully explore all the classes you will need in the game and how exactly they will interact. Their interaction is going to be key in getting this game right.

Added Difficulty: Add sounds or video segments to various aspects of the game.

Title: Pac Man
Difficulty: 5
Description: Develop a basic game of Pac man. The game involves a maze, littered with "pellets" that Pac man must eat. To clear the level, Pac man must eat all the pellets in the maze while also avoiding 3-4 ghosts. Running into a ghost kills Pac man. However, in the maze there are "power pellets" which when eaten allows Pac man to eat ghosts for a limited time. If you are unfamiliar with the game mechanics, be sure to look it up on youtube.com to see how the game is played.

Tips: This game is a very visual game and requires a bit of collision detection (so Pac man won't walk through walls or know when they encounter a ghost). Put together a sample maze and work on the game loop to draw Pac man, the maze and controlling his movements using the arrow keys. Work out the collision detection and see if you can guide Pac Man around a maze efficiently before expanding into introducing ghosts or making the maze bigger.

Added Difficulty: Try making the maze much bigger, double the ghosts and pellets and introduce another player (Ms Pac man?) into the same maze. Work as a team to clear the maze.

Title: Pinball
Difficulty: 10
Description: Create a game that mimics Pinball. The player will start up the game and be presented with a screen full of paddles, widgets, belts, holes, rails etc and be asked to shoot a metal ball through it. The goal is to keep the ball bouncing off all the objects and avoiding the gutter at the bottom of the screen. The ball bounces between various objects on the screen and collects points.

Tips: The trickiest part of this program is determining when the ball hits an object which direction it will go. There is a lot of physics and geometry involved here. To create realism one will have to know the proper amount of drag and what a ball on a board tilted at an angle will behave like. So first thing to do here is collect your formulas and map out which objects will be in the game. How do they effect the ball when hit? Some bumpers speed the ball up, some slow it down (like holes). When a paddle hits the ball coming down at a certain angle, where will the ball go next? Once you collect the equations, you will also need to determine collisions so bring along your collision detection algorithms. Once you have all of these setup, then you can go about drawing your board and creating a class for "Ball". Start with a simple board with one or two objects to see how your equations are working out. Pay special attention to isolating each algorithm into its own function and perhaps creating a library of ball physics to better organize your code.

Added Difficulty: Add sound when items collide or go in the gutter.

Title: Poker With Online Component
Difficulty: 6
Description: Develop a game of poker (Texas hold'em or other of your choice). The user should be able to join the game, receive two cards face down (which they can see) and wager against other opponents. When the betting round is finished, the dealer will put down three cards face up (known as the flop). The goal is to combine the cards in your hand with the "community cards" to form the best hand. After the flop, another round of betting is done. The dealer then puts another card down (The Turn). Another round of betting takes place and then the dealer puts the last card down (The River) to make five cards face up. Another round of betting and then all players show their hand. Winner takes the pot. Build in an online component so that a player can play other players online.

Tips: As with most card games the goal here is to have Card objects with a face value and a suit. Both of these can be done using Enumerations (known as enums in some languages). Cards will be managed in a “Hand” object. In this poker game you may want to also create a “CommunityCards” class to manage the cards which are open to all players. Isolate the classes you need first (Card, Hand, CommunityCards, Players... etc) and then think through the game mechanics. How is a bet going to be made? What if a player raises? Does a player have enough chips? How do you evaluate if a player has a flush or straight or two pair etc? You will need an algorithm to compare two player’s hands to determine who wins. Don’t forget a tie could happen and how to split the pot. Keep in mind how all this would work with players online. As each action happens, we need to notify all players via their Internet connections. Account for timeouts, drops and when a player can actually fold or bet. You wouldn’t want a player to be able to fold or bet out of turn.

Added Difficulty: Add an animation that shows cards flipping over and player’s faces. Make computer players smirk or act panicked as part of a bluff.

Title: Scorched Earth
Difficulty: 5
Description: Make a game that brings back the classic “Scorched Earth”. The game starts by asking the user how many tanks to put onto the field. You need at least 2 players with one of them being the player and the rest can be real players or computer AI players. The game then creates a random landscape of hills and valleys and places the tanks in a somewhat random position, making sure that two players are not on top of one another or right next to one another. The game proceeds to then go to each player and ask them to enter an weapon, angle and power. It then lobs these bombs or missiles at other players in the hopes of destroying them. The game continues until all but one tank is destroyed. Destroying other tanks gives you money that the player can then use to buy more advanced weapons and supplies.

Tips: This game works on some very basic principles including angles and parabolas. Start by sketching out an algorithm that will take an initial angle, velocity (aka power) and starting location and will find all points along an arc taking into account an initial value for gravity and if the bomb hits something in its way (ground or another tank). You will also need a random algorithm for creating the landscape of mountains and valleys and one that can place tanks on this landscape. This means that you may need a function that takes an array of values that form the landscape and use it to place your tanks on it. Think of the landscape itself as a histogram where high values are mountains and low values are valleys. Last, create a standard collision detection function for when you hit another tank. Once you have these things setup, try a two person game for testing.

Added Difficulty: Factor in the effects of wind on your shots. Firing straight into the wind may result in the bomb coming back towards your tank and destroy it.

Title: Slot Machine

Difficulty: 3
Description: Design a game where the user can play a make believe slot machine. The user will be asked to make a wager to play various lines in a 3 x 3 grid. They can play center line, all three horizontal lines, all vertical lines and diagonals. For instance the user can enter $3 dollars and play all three horizontal lines. If the top line hits a winning combination, they earn $1 dollar for that line times the winning combination multiplier. If two cherries and any item pays 2:1, then they would earn $2 ($1 for the line times 2).

Tips: The mention of a grid here should be a dead giveaway that you need a 2D array. You will also need functions that can check a horizontal line, a vertical line and a diagonal. Depending on the number of lines they play, you may need to call all three of these functions one or multiple times to look for winning lines. If they are playing three lines, you would call your horizontal line check function three times... one for the top row, one for the center row and one for the bottom row. Each of these row checking algorithms will then need to look for winning combinations. So it is suggested you keep a list of all combinations and check each row against the list. Once a row has been identified as a winner, you need to look up its multiplier and multiply that by the wager put on that line. The result is then dumped into the player's money total. As for the mechanism to determine what the wheels produce per spin, use a random number generating function.

Added Difficulty: Make it so that rows can have a rocket ship icon which can alter the result. If the rocket ship is pointing up, when the person lands on it it will move that wheel up 1 spot. If the rocket ship is pointing down, it will move the wheel down one slot.

Title: Snake Game
Difficulty: 4
Description: The goal of the snake game is to guide a snake around a square room. The snake starts out small and as it eats a randomly placed fruit, it grows longer. The snake is always moving and as it becomes longer and longer the player has to make sure that the snake itself never touches another part of its body. As soon as the snake touches another part of its body, the game ends.

Tips: Well first you need the square board, you need arrow keys to guide the snake around, you need to randomly place fruit in the game area and a way to keep track of the snake's length. The square board can be the window, the arrow keys can be detected by a type of key listener (java) or key event (.NET), the random fruit will be determined by a random number generator which will choose a random set of coordinates and the snake's length determined by a counter which will be incremented by 1 each time the snake eats a fruit. You will also need a way to determine if the snake's head is at any time touching a square of the board which is currently occupied by its body. To help you do this, think of the board as a grid where each square on this grid is either occupied by a part of the snake (true) or empty (false). If the square currently containing the head of the snake runs into a square which is set to true, we know it hit its own

body. As the end of the snake passes the square, it is setting these squares to false. The head is setting them to true.

Added Difficulty: Introduce obstacles on the board or items they can't run into. Perhaps one of the obstacles is a mongoose and it moves around slowly on the board.

Title: Tic-Tac-Toe with Friend Online
Difficulty: 5
Description: Make a standard game of Tic-Tac-Toe but playable with another player over the Internet or a local network. The game will ask each player their name and whether or not they want to be X or O. A 3 by 3 grid is then displayed and the game asks each player (in turn) to place their mark in one of the empty squares. The goal is to get three in a row. The online component will accept requests from a remote user.

Tips: Tic-tac-toe itself is a rather simple system where you have a board of 3 by 3 squares that you can use an array for. Each square can only be X, O or empty. You need a way for the player to choose one of the squares, validate if it is available, place the mark there if it is and finally look for winning/tie conditions. Each of these steps can be done with its own function and look to keep your design very basic and straightforward. The winning condition check will simply loop through each row looking for three of the same mark, loop through each column and check the two diagonals. Thus you should have some loops which will then supply the rows/columns/diagonals values to a function to check if they match. The online component can be tackled once the game mechanics are in place. When it comes to the remote player's turn, send them the state of the board which will update their display, they put their mark which in turn will send a command back to update your display. Put mechanisms in place in case the remote user drops off so that you can quit the game.

Added Difficulty: Put in drawing functions (if you haven't already) to make the game visual. Try putting in a small chat box so that you can chat with the remote user you are playing.

Title: Type that Word
Difficulty: 5
Description: Develop a typing game where there is a land base at the bottom of the screen and several words the user needs to type fall from the sky. When the user types a word successfully, the base will shoot the word out of the sky. The goal is to prevent any words from falling down and destroying your base. First round will be simple words to type and as each level progresses it will get tougher by making the words longer, introducing new characters and more specialized words. As each letter is typed, make the letters change color in the word so that they know where they are in the typing. Mistakes make the word they are typing start over. Game ends when the base is completely destroyed.

Tips: Build yourself a list of words to form a dictionary. Start with a small set (around 20) at first for testing and then work on an algorithm that will randomly choose one of the words. Then pass

that word to an animation which will make it slowly fall from the top of the screen down. Listen for keypress events, determine the character pressed, look through the words falling and see if the sequence of keystrokes match any of the words. If so, highlight the characters in that word with another color. As soon as the sequence is broken, start over. If the word is typed, delete it from screen. This means that after each character is typed, we need a function which will check the characters typed so far and if they match any word completely. If there is a match, you know to destroy the word. You may also need a mechanism to know when a word has destroyed part of the base and if the base is completely destroyed.

Added Difficulty: Add little explosion animations when a word is destroyed or part of the base has been obliterated.

Check out our website: http://www.coderslexicon.com

Twitter: CodersLexicon
Facebook: https://www.facebook.com/coderslexicon