Dallon Jarman
Assignment one
CYB136

For the program I created a mock bank called Bank of Gallon (nickname of mine). The purpose of this program is to allow the user to create multiple bank accounts for multiple users such as friends or family members. When the user loads up the program it asks you for the account name. From there you are allowed to put in as many words as you would like to name your account. If you wanted the account to be called "Dallon Jarman's Account." You can do so. If you just want to account to be called "Checking" you can also do that. After you input the name, you can deposit as much money as you would like. You can also input cents as well. After you create your initial account, it will give you more actions to do. The other actions are as followed, deposit to an open account, withdrawal money from an open account, create another account, or view all open accounts. Used classes and everything else I have learned from the chapters in this program. Went through removed any obvious flaws that can allow the user to break the program. The user is not allowed to deposit or withdrawal a negative amount of money. The user can only input the numbers 1-5 when selecting their next action. When creating another account. The user may add spaces in the account. The user may do anything they want. As long as it is within what the program will let them. If the user makes an error the program will tell them and will tell them how to fix it. I have tested everything I can think of to make this program error proof. I can't promise this program still has hundreds of bugs. It is good enough to where a typical user will struggle to break it.

This program has a bug when withdrawing money. The bug that could be bad is when the user wants to withdrawal money from their account. The program will allow the user to select from accounts that do not exist. The program does not show the new account created or associate a name to the account. It is an account that nobody can see unless you know the account number. This can be bad because if an account that nobody knows about except for the individual with the account number then money can go in and out of the account undetected. This account can also cause errors when you do create an account with the same number as the hidden account. Because now this hidden account is becoming known. The user with the new account can either have millions of dollars or the account with millions of dollars could reset to zero. This is even worse. An account with millions going to zero. That would cause so much missing money. Banks would have all this extra money and would need to explain where all this money went. People who created these hidden accounts would lose millions of dollars and would cause chaos. A security flaw like this would be bad.

The language features its misusing is vectors. I am still learning about vectors and how they work on a much more in-depth level. When you input any random number, it creates a new part of the vector. The program will not tell you this vector exists unless you specifically know the correct number. I do not fully understand how vectors work but I do know that a flaw like this would be bad on a wide scale use. It is important for developers to evaluate everything and look outside the box to really evaluate and exploit their code before releasing it on a wide scale.

There are millions of companies that exist. Every company with their own program and software. Usually, most companies will rely on a bigger company for their software. When millions of customers using one software. That causes many unexpected issues and problems to arise. You can do so much testing and push things to their limits. Once software hits the mass markets that will really bring certain and unexpected bugs and issues to light. This also explains why it is a bad idea to buy products on day one. Usually, these products will have issues the company never expected. This is also great for the development of the company because now the know what will break and how to protect against it in future released.

A bug like this is hard to detect when looking over the code. You will not think much of it, and I did not until I had some friends evaluate out the code and bring it to my attention. This can cause so many flaws. I had so many friends evaluate the code over and over until one of my friends was able to exploit this bug and find the flaw. From there I did more testing and found how I can fix it. This is hard to find. Any bug is not easy to find and that is why it is called a bug. The program works perfectly but any step outside of the line causes loads of problems. Many developers will look over bugs because nobody expects the unexpected. Evaluating your software to its fullest it great but also releasing it to the public to test is even better. People will be able to find loopholes and back doors. Test various parts of your code you never expected. Customers will be able to stick their foot in places with no idea how they got there.

Finally, bugs in code are bad. They are extremely difficult to find. Most bugs are easy to find and will usually be removed from the code instantly. While other bugs require hours of testing and hundreds of people to test in order to find. No code is flawless and will always have exploits. These exploits can be extremely difficult to find and companies are always patching possible expoist.