



Learning Experience Report

Learning something new, is never what we could call an easy experience; but that is the most rewarding part of it, the opportunity to overcome a new challenge. Learning the Java programming language was no different. Each class represented a whole new way to interpret a concept we thought we knew, and in some way, a new obstacle. I had received an introduction to programming to C++ last term in Miami Dade College. Also, I had received some classes about programming, using Java as a programming language, in UCI (Universidad de las Ciencias Informáticas) in Cuba. Even so, I found myself challenged once more, but having a teacher willingly to walk this path with us, made possible to enjoy the experience.

The reason for me to walk this path comes from my childhood. When I was younger, about 7 or 8 years old, I used to go to the computer lab, and sneak in during classes for older students. They were just working on Microsoft Office, but at the time, they were doing things I did not know how to do. The most deciding moment was during the final project of a group of students. They had a word document, with a hyperlink to a PowerPoint illustrating the explanation they had given to a topic. I don’t even remember the topic they were discussing; my focus was on the action I had just seen; I couldn’t imagine how they managed to make a PowerPoint presentation being launched from a word document. In that moment it struck me; I wanted to be able to tell the computer how to behave.

This story might not seem important to this topic, but the reason why I am writing this, and why I completed, or will soon complete, this class; was that story. I was reminded of it during the class. With each chapter I was a step closer to what could have been just a child’s dream. The class taught me how it is possible to “tell the computer what to do”. I learned that there are still steps after I write a code, steps that make possible for the computer to understand what I mean with my code. I learned what an algorithm is, and why is it needed to “communicate” with the computer.

One of the most interesting parts for me was to learn to understand a problem before attempting to write any code. Before that I only focused on the code itself, I never stopped to think about the analysis needed before getting there. This was the first class where I got to see this perspective. Not only that, but we were encouraged to go beyond just merely answering the exercise, to ask, “what if” and add a new condition that would change the approach made before.

Also, we learned why it is called OOP (Object Programing Programing), and why it was so important. We learned that we can treat, pretty much, everything we need as an object when programming. The important property of this feature is that each object “knows” its attributes and methods, allowing the programmer to write the code just once in the object. Afterwards is only needed to ask the object to execute a certain method already defined. On this topic we learned about classes, which contains the definition of the methods and fields. We also learned about objects as instances of the already defined classes.

We also improved our understanding about arrays and learned about ArrayList, which represented an improved way to handle many elements. We faced more complex exercises which made us develop a better understanding of usual conditional statements and loops. Learning to work with files was both difficult and interesting, providing us a new sight to new options for future projects.

Exception handling was a very difficult yet rewarding topic to learn. It allowed us to take more control over what happened in our program; a way to make the program end in our own terms. This was probably the topic that inspired me the most because the computer had to follow the instructions more than before. We were able to control better the possible mistakes, and possible unwanted inputs from the user.

Another important topic was inheritance. It was rather difficult to understand in the beginning. How is one code supposed to inherit another code? That was probably the first question for most of us. But as Professor Hien continued explaining, the subject became clearer, as well as its usefulness to create more clean, readable and optimized code. The concept of abstract class and interfaces were also very complex themes. As the name itself states, it was very abstract and hard to visualize. Luckily as time went on, we were also able to understand this part and all properties tied to it. An example was going back to herency. The status of being “abstract” is inherited from a class to its children. Another, that a class that has one abstract method, must be declared as abstract, and this method must be implemented on its child. Those were the ones that stayed the most in my memory at the beginning. Once I started working on the exercises, one more property made sure to leave trace: an abstract class cannot be instantiated. I will never forget myself staring at my code for an exercise, wondering what was wrong; until I understood that I was trying to create an instance from an abstract class.

The final topic for this class was recursion. This is an interesting one, but different from the ones before. With recursion the theory was easy to understand, but the implementation and use, not so much. We could understand the idea of using recursion until a certain condition is met. Break, in smaller parts, a problem, until you can easily answer it. Same way series work on math, and other examples even in real life. The hard part was, we never had to stop and think, how is it done. When coding, we need the answer to that question, because we need to explain it to the computer step by step.

When we started with Graphical User Interface it seemed a whole new world. We could not imagine how we could put together everything we had learned up to this moment. Being able to create a visual representation of our program looked like a distant and impossible task. Setting up the tools we needed was really hard work and needed some teamwork. Each of us was able to provide information for this purpose. After setting up the tools, each of us needed to work alone to make the final project possible.

Working in the final project was an experience I will never forget. It allowed me to use all skills we were learning up to this point. First, I needed to understand my problem, what I was asked for. Then it was time to design the UML, it might not be the most fun part, but it is definitely an important step to develop code. It allowed me to obtain some perspective over what it needed to be written to obtain the results I wanted. So, it was time to code. I moved to develop my classes according to what was explained. I used my own notes from class, the book and some extra articles, to cover specific topics I wanted to include in my code. Line by line, I started working on my program. Once the classes were done, it was time to work on the driver class and the GUI. I already had an idea for the first one; the second one, was still new. First, I did a few simple designs on how I wanted it to look like; then I used again the book, notes and everything I could find to try to get as close as I could to my original design. Then, I was able to focus on the details, and think of an answer for possible entries from the user. I tried to use as many exceptions as I could to stop my code from ending abruptly. It was hard work, and I was facing all topics we had received altogether. But then again, having layered out everything beforehand, allowed me to look for exactly what I needed. Now what was missing was test subjects, and that is where my friends and family came into play. They tried my program over and over, every time I corrected something, they would gladly try again and even gave me advice on grammar and structure. Working on this project made me realize that I was able to do much more than what I thought I was capable of. Working on my own, using knowledge I had gathered from my time in this class. It might still not be perfect, and I know there is still much to work and learn, but I was able to work by myself.

Being able to participate in this class was an unforgettable experience. It did not just mean a challenge full of surprises and learning opportunities, but also gave me the opportunity to meet new persons and get to experience a whole new virtual world. I was able to regain confidence in the topics I needed to study, and in myself. I met classmates that made this experience a lot more enjoyable. People to talk to and have fun sometimes, but also ,that proved to be ready to help whenever they could. Students like me, taking this battle one step at a time. We all gave the best we could, students and our teacher alike. We had the privilege of receiving class from someone that enjoys what he does. A professor that loves to be able to teach us the tools to be able to conquer this amazing world. A teacher that encouraged us to break our own barriers, to go one step at a time and understand how to learn from our mistakes. This is a journey that just began, we still have a way ahead of ourselves. Luckily, we have the tools to embrace what the future holds for us. What else can I say? I really enjoyed the time as spent on this class. I leave this class feeling that I’m a step closer to complete the dream that started when I was just a little kid; with the skills and tools to keep going until I see it become a reality.