B. THE PROGRAMMER'S MIND

All programming languages require that we take a certain logical approach to solving problems, and the logical structures we use in all programming languages are the same, so when I teach a beginning prog. class in any language, certain topics will be the same. If you have previously taken a first-semester prog. class, you will be getting a review of these topics in this class.

I'm going to take a fairly traditional, tried and true approach to teaching programming in this class. This traditional approach will help you to develop a "programmer's mind". The way a prog. thinks about solving problems is different from the way most people think, or you could say programming uses a different part of your brain, and a set of talents that you may not have used before, depending on the kind of work you do. I think it is a very enjoyable activity. It is about solving problems. If you like solving puzzles, or setting your mind to the solution of a complex problem, and figuring out the steps to solving it, you're going to enjoy programming.

I'd like to talk about the qualities you find in good programmers. These may be qualities which you already have. If not, you can develop these characteristics with some mindfulness and effort.

1) A programmer is someone who, as I mentioned before, enjoys solving problems, and who believes that problems, no matter how complex, are solvable. You have to believe that if you chip away at a problem piece by piece, you will in the end, complete the solution, because it is true! A programmer friend of mine says a programmer is an “optimistic bulldog”.
2) You need patience, and perseverance.
3) You have to have a sense of humor about the work you are doing, or at least a sense of ironic distance from the frustration of the work you are doing, because it can be frustrating. You have to have a high tolerance for frustration.
4) You must be methodical. If something doesn’t work, you must try something else. And if that doesn't work, you must try something else. Be willing to try every avenue, every possibility to find a solution.
5) Programming is very creative work. As a programmer, you must develop your intuition to help you discover solutions to programming problems. The art of programming.
6) A programmer must have an eye for detail. You can't be sloppy, because programming languages are not like natural languages. Natural languages are ones that people speak, such as English, Spanish, or Chinese. When we speak to each other in our natural language, we don't always speak precisely. Our listeners can fill in the blanks in our sentences, and understand us. We have what is called "a ground of knowledge" which we both share, which allows us to interpret correctly what the other person means. “Please sit down”

So programming languages are not like natural languages. They are very precise in the way you must write the statements. A statement is a sentence in the language. The rules you follow when writing statements are called the syntax of the language. Syntax mean grammar. Programming languages allow you to communicate with a machine, not a human being. And computers are not as smart as humans. They can't fill in the blanks. So programming languages have very strict syntax rules, and you can say that the computer "understands" these rules, but nothing more. For example, if you said to me, "change color of your shoes", I would understand what you meant, even though your sentence is ungrammatical. On the other hand, if you want to draw a red oval in an applet window, you might write these two Java program statements:

```java
    g.setColor(Color.RED);
    g.drawOval(20, 30, 50, 20);
```

and those statements will be "understood". But if you left out the `g.` or the `semicolons`, you will get a syntax error message from the Java compiler. So you need an eye for detail to spot your program errors.

8) A programmer must think logically to solve a problem.

In terms of computer programming, "solving a problem" means getting the computer to perform one or more tasks for you. The "problem" is the work that you want the computer to do. For example, the problem in the third lab assignment is to find the largest of 10 numbers input by the user.

A programmer will take the statement of a problem("find the largest number") and apply logic to solve the problem. Solving the problem means designing a program to perform the task or tasks outlined in the problem statement. It means breaking down the task into a sequential series of logical steps. This set of steps is called an **algorithm**

In analysing this problem, we can break it down into these steps,

1) ask the user to enter 10 numbers (probably ask for one at a time)

2) find the largest
3) display the largest back to the user

So if the user types in this sequence of numbers - 12, 34, 3, 6, 89, ......2, 4, our program will report that 89 is the largest. Remember, we don't know in advance exactly what numbers the user will type!