Basic was originally designed at Dartmouth University by John Kemeny and Thomas Kurtz. These men also developed compilers but things such as Fortran and ALGOL 60. As this university was a liberal arts college, they wanted to make a language that was easy for nonscience students to understand, pleasant, fast turnaround, free and private access, and user time friendly.

The original Basic was small and not very interactive. There was no way for a program to get input from a user. Programs had to be built and then ran. No intermediate way to change the program while executing was available. It originally only had 14 data types. It was limited and easy to learn.

The most important part of the program was that it was widely used through terminals connected to a computer. Lisp was also used but was not as popular in the 1960s. Before terminals, computer information was punched in on cards or tape. Much of the design came from Fortran and ALGOL. Although, there was no standardization of the syntax. Huge corporations used a more complex form (basic-PLUS) to develop things such as minicomputers in the 1970s.

Basic was known for its poor structure of programs and many other flaws. In later years, VB was developed and had much more praise. We discuss this more in HW #2.

I think that without the comments on the code in the book it would not be so easy to implement. As I have never seen basic before I do not know any of the syntax. The comments basically explained everything the code was doing. Although I think perhaps it would be easier than some languages. I have spent many hours reading foreign code and changing small things to figure out what it was doing. Basic, having such limited data points, I would assume would be fairly easy to understand if I had to without comments. It would definitely take some googling or a syntax table but I think it would be easy to understand. Implementation could be considered as easy or hard. To complete the task would be easier because we have so many more data types and things to use in other languages. But working around such basic code to make a nice user interface would be challenging. For example, in my code I have to make loops for the user to decide if they want to add more integers. I would have no idea how basic did that before. From the reading I assume the user had to manually input all the data points before running the program. When in my code the user can enter data as they please.

Here is my interpretation of the basic code. I know I got the goal of the code done, but I did it in my own user friendly way as Cooper would have wanted.

public class basic{

public static void main(String args[]) {

ArrayList<Integer> intlist = new ArrayList<Integer>();

int listlen, average, result = 0;

int counter = 0;

int sum = 0;

//read input into array and computer average

Scanner sc = new Scanner(System.in);

String end = "y";

while (end.equalsIgnoreCase("y")){

System.out.println("Please enter integers between 1 and 99 to add to the array");

listlen = sc.nextInt();

if (listlen < 100 && listlen > 0){

counter ++;

Sum ++;

intlist.add(listlen);

System.out.println("Add another integer? (Y)es (N)o");

end = sc.next();

}

else{

break;

}

}

//computer the average

average = sum/counter;

for(int i = 0; i < intlist.size(); i++){

if (intlist.get(i) > average){

result = result + 1;

}

}

System.out.println("The number of values above the average is " + (result-1));

}

}

