/**
 * This program prints my name, age, major, and an ASCII star
 * @author Cody Narber
 * @version 8-26-07
 */

public class Proj1 {
    public static void main(String[] args) {
        System.out.println("Name: Cody Narber");
        System.out.println("Age: 23");
        System.out.println("Major: Computer Science");
        System.out.println("\n");
        System.out.println("Here is my star: ");
        System.out.println("\n");
        System.out.println("        *        ");
        System.out.println("  *     *     *  ");
        System.out.println("    *   *   *    ");
        System.out.println("      * * *      ");
        System.out.println("* * * * * * * * *");
        System.out.println("      * * *      ");
        System.out.println("    *   *   *    ");
        System.out.println("  *     *     *  ");
        System.out.println("        *        ");
    }
}
Key Points of Interest:

- Variables
- Data Types
- Arithmetic Operations
- Input
- Random Integers
- If-Statements (boolean logic)
Variables denote blocks in memory that allow us (the programmers) to store and recall information at any point in the program. In order to create a variable we will need to tell the computer what kind of information can be stored in it (whether it be text, numbers, etc.) This is known as the type:

```java
int x;
int y;
String text;
```

The above lines of code created 3 blocks in memory the first two blocks will hold integers and the last will hold a string of text (we printed out strings in the first assignment).

This is known as declaring a variable.
Below are a listing of some common data types that we will be using throughout the entire semester:

- **int**  - integers,  
  ex: -2, 0, 15, -36, 212
- **double**  - decimal numbers,  
  ex: -2.3, 1.0, 0, 100.65
- **char**  - letter/symbol  
  ex: 'a', 'S', '+', '?', '&'
- **boolean**  - a boolean value  
  [true, false]
- **String**  - sequence of characters  
  ex: “Hi There”, “a tree”
Declaring is done only once. This sets up the block in memory. Whereas, assignment is when you put a value into that block. You can assign a new value as many times as you want.

```java
int x; //declaring a memory block called x for integers.
x = 5; //assigning a value 5 into that memory block.
x = 10; //assigning a new value into x.
char let; //declaring a character variable.
let = '%'; //storing the percent sign into the variable let.
String word; //declaring a String variable.
word = "Test"; //assigning the word Test into the variable word.
```
Below are math operators that are used with numerical types (int, double)

- +  addition
- -  subtraction
- *  multiplication
- /  division
- %  modulus (only used with integers)

//given x and y are variables with 5 and 10 assigned respectively
int z = x + y; //assigns the sum of x and y into a new variable z (15)
int w = x * y; //assigns the product of x and y into a new variable w(50)

z = x+(y*z);
int r = z / w;
In order to get input from the user there are a few things we need to add to our code.

1) `import java.io.*;` //add this line at the very beginning of your code, above the class header even

2) `public static void main(String [] args) throws IOException` //Change your main method so that java knows that you are using user input.

3) `BufferedReader r = new BufferedReader(new InputStreamReader(System.in));` //add this line right inside the main method //r will be used to read the user's input.
Doing the three things mentioned will allow us to wait for user input and use it in our programs. Making our programs much more dynamic.

Inside the main method use this line of code to ask for user input:

```java
r.readLine(); //use if your BufferedReader is named r
```

This will pause your code and wait for the user to type something and press enter. Once the user presses enter, what was typed on that line is read-in as a String and should be stored, so you will generally use this line of code to read-in the user's text and store it:

```java
String input = r.readLine();
```
import java.io.*;

public class TestInput
{
    public static void main (String []args) throws IOException
    {
        BufferedReader r = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter your Name: ");
        String name = r.readLine();

        System.out.print("Enter your Hobby:");
        String hobby = r.readLine();

        System.out.println("Greetings "+ name + " one who does "+ hobby);
    }
}
If you want to convert the String that is read you can use the following code:

```java
int num1 = Integer.parseInt(r.readLine());
    //do this for integer input

double num2 = Double.parseDouble(r.readLine());
    //do this for decimal input
```
Sample Program of getting the user's number, squaring it, and displaying it back to the screen.

```java
import java.io.*;

public class CovertInt {
    public static void main (String []args) throws IOException {
        BufferedReader r = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter your Number: ");
        String snum = r.readLine();
        int num = Integer.parseInt(snum);

        int sqnum = num*num; //squaring it.

        System.out.println(num + " squared is "+sqnum);
    }
}
```
An if-statement is used to control the program's flow and only execute a block of code when a certain condition is true. Below is the basic syntax for the if-statement.

```java
if(<condition>)
{
   <block of code>
}
```

There is also an if-else statement in which there are two blocks of code, the first is run when the condition is true and the second block when it is false:

```java
if(<condition>)
{
   <1st block of code>
} else {
   <2nd block of code>
}
```
Condition Operators used for comparing

- ==  equals (be careful not to confuse with the assignment)
- >   greater-than
- >=  greater-than or equal to
- <   less-than
- <=  less-than or equal to
- !=  not equal to
- ||  or
- &&  and
Write a program that will ask the user for two decimal numbers, then list a set of operations for them to choose (1 – sum, 2 – difference, 3 – product, 4 – quotient). Finally display the solution to their choice. The Terminal is as shown below:
import java.io.*;

public class TestIf
{
    public static void main (String []args) throws IOException
    {
        BufferedReader r = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter number 1: ");
        double num1 = Double.parseDouble(r.readLine());
        System.out.print("Enter number 2: ");
        double num2 = Double.parseDouble(r.readLine());

        System.out.println();
        System.out.println("1-Sum");
        System.out.println("2-Difference");
        System.out.println("3-Product");
        System.out.println("4-Quotient");
        System.out.print("Choose your operation by typing"+" that number: ");
        int choice = Integer.parseInt(r.readLine());

        //...continued on next slide
//int choice = Integer.parseInt(r.readLine()); ...

double sol = 0;
if (choice == 1) {
    sol = num1 + num2;
}
if (choice == 2) {
    sol = num1 - num2;
}
if (choice == 3) {
    sol = num1 * num2;
}
if (choice == 4) {
    sol = num1 / num2;
}

System.out.println("The solution is: "+sol);
}//closes main
}//closes the class
//picking up from the first part of the solution

int choice = Integer.parseInt(r.readLine()); ...

int sol = 0;
if (choice == 1) {
    sol = num1 + num2;
} else if (choice == 2) {
    sol = num1 - num2;
} else if (choice == 3) {
    sol = num1 * num2;
} else {
    sol = num1 / num2;
}

System.out.println("The solution is: "+sol);
} //closes main
} //closes the class
Random Numbers

In order to get input from the user there are a few things we need to add to our code.

1) import java.util.*;  //add this line at the very beginning of code, above the class header, just like with java.io.*

2) Random rand = new Random();  
   //add this line right inside the main method  
   //rand will be used to generate a random  
   //number, just like BufferedReader
Random Numbers

Doing the two things mentioned will allow us to generate a random number (integer).

Inside the main method use the line of code below to generate a random integer:

```
rand.nextInt(<integer>)  //use if your Random is named rand
//<integer> is a placeholder for a
//number
```

This will generate a random number from 0 (including) up to but not including <integer>.

EX: 
```
rand.nextInt(10);        //generates a number in the range [0,9]
rand.nextInt(4);         //generates a number in the range [0,3]
```
import java.io.*;
import java.util.*;

public class TestRandom {
    public static void main (String [] args) throws IOException {
        BufferedReader r = new BufferedReader(new InputStreamReader(System.in));
        Random rand = new Random();

        System.out.print("What is the upper bound for the num: ");
        int upper = Integer.parseInt(r.readLine());

        int randNum = rand.nextInt(upper);

        System.out.println("The random number is: "+randNum);
    }
}
Test Your Skills

Write a program that generates a random number from 1-10 and asks the user to guess it. Then using if-statements display whether the user was high, low, or dead-on. Also display the user's guess, and the actual

Tip: break the program into steps, and code each step one at a time.

Example of output of the finished running program:
To run your program in lab, go into the BlueJ Project Directory that you created you should see something like this:

Right-Click go to “New->Text Document” call it “Run.bat”. Then Right-Click that file and choose “Edit”. In the text editor type:

```
java <your class name>
pause
```

where `<your class name>` is the name of the class you created in BlueJ. Save and close. Now you can just double-click run.bat to run your project.