JAVA VARIABLES

Programs work by manipulating data placed in memory. The data can be numbers, text, objects, printers to other memory areas, and more besides. The data is given a name so that it can be recalled whenever it is needed. The name and the value of the data is known as a variable. We have several kinds of variables in Java dependent on the kind of value that is stored in it.

Number variables

To store a number in Java, you have lots of options. Whole numbers such as 8, 10, 12, etc, are stored using the '`int`' variable. The `int` stands for integer. Floating point numbers like 8.4, 10.5, 12.8, etc., are store using the '`double`' variable. You do the storing with an equals sign (=). For example:

```
public static void main (String [] args) {
      int first_number;
      first_number=10;
      System.out.println ('My First Project');
}
```

To tell Java that you want to store a whole number or integer, you first type the keywords ‘int’, followed by a space. Then you type the name you want for the integer variable. In the above example, the name of the integer variable is first_number.

To store something in the variable called first_number, you type an equals sign and then the value you want to store. In the above example, we store the value of ten (10) in the integer variable we’ve called first_number.

In naming of variables, there are some rules

- Variable names don’t start with a number although you can have numbers elsewhere in the variable name.
- Variable names can’t be the same as Java keywords. E.g. int is a Java keyword and can’t be used to name a variable.
- You can’t have spaces in your variable names. Usually, underscore (_) can be used or instead

the second word should begin with a capital letter.

- Variable names are case sensitive. Therefore, firstNumber and FirstNumber are different variable names.

The double variable is also used to hold floating point values. A floating point value is one like 8.7, 12.5, 10.1. In other words, it has a "point something" at the end. If you try to store a floating point value in an int variable, NetBeans will underline the faulty line. If you try to run the program, the compiler will throw up an error message.

Change the int in the previous example to double and set its value to 10.5.

```
public static void main (String [] args) {
      double first_number;
      first_number=10.5;
      System.out.println (first_number);
}
```

Two more variable types you can use are 'short' and 'float.' The short variable type is used to store

smaller number, and its range is between +32, 768 and -32, 767. We can use short in place of int although int store larger numbers, viz: ±2147483648.

Double can store very big numbers of the floating point variety. Instead of using double, float can be used if the decimal number to be stored is not so large. When using float, you need to always type the letter 'f' at the end of the value before the semicolon. Like this:

```
float first_number=10.5f;
```

String Variables

Variable can hold text just as they do hold numbers. You can store just one character, or lots of characters. To store just one character, the '`char`' variable is used. Usually, you'll want to store more than one character; to do so you need the '`string`' variable type.

To set up a string variable, you type the word '`String`' followed by a name for your variable. Note that there's an uppercase 'S' for String. Again, a semicolon ends the line.

```
String first_name;
```

Assign a value to your new string variable by typing an equals sign. After the equals sign the text you want to store goes between two sets of *double quotes*.

```
first_name= “Williams”;
```

If you are storing just a single character, then the variable you need is `char` (lowercase ‘c’). To store the character, you use *single quotes* instead of double quotes. For example:

```
char first_name= ‘W’;
```

If you try to surround a char variable with double quotes, NetBeans will underline the offending code in red, giving you “incompatible type” errors. You can however, have a String variable with just a single character. But you need *double quotes*. For example:

```
String first_name= “W”;
```

Boolean Variables

A `Boolean` value is one with two choices: true or false, yes or no, 1 or 0. In Java, there is the Boolean variable for Boolean values; It is written like this:

```
boolean user= true;
```