**Name**

**Programming II**

**Lab Exercise 6.6.2025**

In the following exercises you will experiment with different data types. Put your answers to the questions in source code comments.

**Exercise 1**

The following program uses primitive data type short:

class ShortEg

{

public static void main ( String[] args )

{

short value = 32;

System.out.println("A short: " + value);

}

}

The word "value" in this program is a variable---a name for a section of memory that holds data using a particular data type. In this case "value" will be the name for 16 bits of main memory that uses short to represent an integer. (There is much more about variables in the rest of these notes.) This program puts the value 32 into "value." Then it writes out:

A short: 32

In other words, the next line of the program examines the variable and writes out what it finds.

**Your Job**  
Create a file called ShortEg.java. Compile and run it. Check what it writes onto the screen. Now edit the program so that the 32 is changed to some other small number, say 356. Compile and run the program. Everything should be fine.

Next change the number to 35000 and try to compile and run the program. This number is too large to work with the data scheme short (in other words, it cannot be represented in 16 bits using data type short.) What happens?

Now edit the program (don't change the 35000) so that the data type is int. Compile and run the program. Is there a difference?

**Exercise 2**

The following program uses primitive data type double:

class DoubleEg

{

public static void main ( String[] args )

{

double value = 32;

System.out.println("A double: " + value);

}

}

Compile and run the program. Does its output (what it puts on the screen) differ from the output of the first program in the previous exercise? In this program, "value" is the name for 64 bits that uses the double data type to represent floating point numbers.

It is perfectly OK to use the variable name "value" for both programs. The use of names like this is a way for us to describe what we want to happen in the computer; it does not permanently reserve part of computer memory for any particular use.

Now try to "break the system." Look in the chapter at the chart of primitive data types and their allowed ranges of value. Change the "32" to a value that is too big for double. You may wish to use scientific notation for this.

**Exercise 3**

The following program also primitive data type double. This program computes and writes out the value of exp( 32 ). This is the base of natural logarithms "e" raised to the power 32. (Don't worry much about this. The point of the program is not the math but the floating point numbers.)

class DoubleCrash

{

public static void main ( String[] args )

{

double value = 32;

System.out.println("e to the power value: " + Math.exp( value ) );

}

}

Compile and run the program. Does it compile and run correctly? Now change the 32 to larger and larger numbers until something goes wrong.

**Exercise 4**

The following program uses primitive data type char:

class CharEg

{

public static void main ( String[] args )

{

char ch = 'A' ;

System.out.println("A char: " + ch );

}

}

The variable "ch" is 16 bits of main memory that uses a scheme for representing characters. The character 'A' has be placed in it. The program will write:

A char: A

Do the following:

* Change the 'A' into 'Z' and compile and run.
* Change the 'A' into 'AA' and try to compile the program.
* Change the 'A' into ' ' and compile and run the program.
  + Notice carefully: there is a single space between the two ' marks.
* Change the 'A' into '' and try to compile.
  + Notice carefully: there is no character between the two ' marks.
* Change the 'A' into "A" and try to compile the program.
  + (The double quotes " signify a String, which is something different from a char).

When you have completed these four exercises, you are to attach your commented final source code for each to this sheet and turn in.