RUSS SEAMAN

Assignment 4 – JAVA TYPE ASSIGNMENT

1. Byte:
   1. Byte cannot be stored as any other type because of its small size. A byte is 8 bits and the other types are 16 bits +, so physically it will not fit into the allocated space.
2. INT:
   1. Int can store a Byte, Short, or Char. This is because the size of them is smaller than the allocated space an int takes up(32 bit). Int also cannot take numbers that include a decimal.
   2. The rest of the primitives cannot be stored as an int.
3. Double:
   1. Double can store Byte, Int, Short, Long, Float, Char. This is because the size of a double is 64 bits, which means anything smaller, it can store the values of.
   2. Double cannot store values of Boolean or String. Boolean values are true or false and String is an object.
4. Short:
   1. Short can only be stored as Byte, because Byte is the only thing smaller than Short.
5. Long:
   1. Long can be stored as Byte, Int, Short, and Char. A long is (64 bits), which means that anything smaller can be stored.
   2. The exception to the size rule comes in at the decimal and float, which a Long cannot store a decimal or a float, even though its smaller.
6. Float:
   1. A float can store a Byte, Int, Short, Long and Char. A Float is 32 bits and the listed types are all smaller than it. Therefore, it can be assigned.
7. Char:
   1. Nothing be stored as a char; this is because the other types are signed whereas a char is unsigned. This means that a byte, for instance, the first bit represents the sign (+/-) of the number. A character will not have this.
8. Boolean:
   1. Boolean is looking for a value of true and false, which no other data type would give. This means that no other data type can be stored as a Boolean.
9. String:
   1. No values can be stored as a string without type casting. This is because java treats string as an object.

public class Assign4{

public static void main(String[] args) {

byte BYTE = 0;

int INT = 0;

double DOUBLE = 0.0d;

short SHORT = 0;

long LONG = 0L;

float FLOAT = 0.0f;

char CHAR = 0;

boolean BOOLEAN = false;

/\*---OBJECTS CLASS-- \*/

String STRING = null;

Coke coke = new Coke("cherry");

Coke coke2 = new Coke("cherry");

String result = coke.equals(coke2) ? "EQUAL" : "NOT EQUAL";

System.out.println(result);

// BYTE COMPARE

// BYTE = INT; ERROR

// BYTE = DOUBLE; ERROR

// BYTE = SHORT; ERROR

// BYTE = LONG; ERROR

// BYTE = FLOAT; ERROR

// BYTE = CHAR; ERROR

// BYTE = BOOLEAN; ERROR

// BYTE = STRING; ERROR

//INT COMPARE

INT = BYTE;

// INT = DOUBLE; ERROR

INT = SHORT;

// INT = LONG; ERROR

// INT = FLOAT; ERROR

INT = CHAR;

// INT = BOOLEAN; ERROR

// INT = STRING; ERROR

//DOUBLE COMPARE

DOUBLE = BYTE;

DOUBLE = INT;

DOUBLE = SHORT;

DOUBLE = LONG;

DOUBLE = FLOAT;

DOUBLE = CHAR;

// DOUBLE = BOOLEAN; ERROR

// DOUBLE = STRING; ERROR

//SHORT COMPARE

SHORT = BYTE;

// SHORT = DOUBLE; ERROR

// SHORT = INT; ERROR

// SHORT = LONG; ERROR

// SHORT = FLOAT; ERROR

// SHORT = CHAR; ERROR

// SHORT = BOOLEAN; ERROR

// SHORT = STRING; ERROR

//LONG COMPARE

LONG = BYTE;

// LONG = DOUBLE; ERROR

LONG = INT;

LONG = SHORT;

// LONG = FLOAT; ERROR

LONG = CHAR;

// LONG = BOOLEAN; ERROR

// LONG = STRING; ERROR

//FLOAT COMPARE

FLOAT = BYTE;

// FLOAT = DOUBLE; ERROR

FLOAT = INT;

FLOAT = SHORT;

FLOAT = LONG;

FLOAT = CHAR;

// FLOAT = BOOLEAN; ERROR

// FLOAT = STRING; ERROR

//CHAR COMPARE

// CHAR = BYTE; ERROR

// CHAR = DOUBLE; ERROR

// CHAR = INT; ERROR

// CHAR = SHORT; ERROR

// CHAR = LONG; ERROR

// CHAR = FLOAT; ERROR

// CHAR = BOOLEAN; ERROR

// CHAR = STRING; ERROR

//BOOLEAN COMPARE

// BOOLEAN = BYTE; ERROR

// BOOLEAN = DOUBLE;ERROR

// BOOLEAN = INT; ERROR

// BOOLEAN = SHORT; ERROR

// BOOLEAN = LONG; ERROR

// BOOLEAN = FLOAT; ERROR

// BOOLEAN = CHAR; ERROR

// BOOLEAN = STRING;ERROR

//STRING COMPARE

// STRING = BYTE; ERROR

// STRING = DOUBLE; ERROR

// STRING = INT; ERROR

// STRING = SHORT; ERROR

// STRING = LONG; ERROR

// STRING = FLOAT; ERROR

// STRING = CHAR; ERROR

// STRING = BOOLEAN;ERROR

}

}

public class Coke{

String flavor;

public Coke(String flavor){

this.flavor = flavor;

}

public void setFlavor(String flavor){

this.flavor = flavor;

}

public String getFlavor(){

return flavor;

}

@Override

public String toString(){

return("Flavor: " + flavor);

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null || obj.getClass() != this.getClass())

return false;

Coke coke = (Coke) obj;

return (coke.flavor.equals(this.flavor));

}

}