# Java Cheat Sheet: reminders about Java syntax and other details for CISC 101
# Use As An Example Only: Add, Remove, Customize/Optimize for You

**Variable Names:**
```
   studentName
   middle_initial
   student5
   GST_RATE
```
upper & lower case matters, all UPPER for constants

**Primitive Types:**
```
   int
   double
   char
   boolean
```

**Declarations & Assignments:**
```
   int x;
   x = 14;
   double d = 15.2;
```

**Arithmetic Operators:**
```
   +, -, *, /, %
```

**Relational Operators** (primitive types only):
```
   <  >    <=    >=    ==    !=
```
(Note comparison for equality is double equals)

**Logical Operators:**
```
   &&, ||, !
```

**Increment/Decrement**:
```
   x++; // means x = x + 1;
   x--; // means x = x – 1;
```

**Assignment Operators:**
```
   x += 3; // means x = x + 3;
   x -= 7; // means x = x – 7;
   similarly for *=, /=, %=
```

**Comments:**
```
   // rest of line
   /*
      multi-line
   */
```

**Arrays:**
```
   // declare & create array of 10 doubles
   double[] arr = new double[10];
   number of elements in arr: arr.length
   // declare, create, initialize 1D array
   int[] a = {1, 5, 2, -3};
   // create 2-D array of ints
   // with 3 rows, 4 columns
   int[][] table = new int[3][4];
   number of rows in table: table.length
   colums in row r: table[r].length
   // fill this in with access patterns,
   // smallest in array, largest
   // index of location versus contents
```

**Strings:**
```
   String s = "abc" + "def" + 13;
      // s gets "abcdef13"
   int len = s.length(); // len gets 8
   if (s.equals(t))
      // true if t exactly the same as s
   int x = s.compareTo(t);
      // 0 if they're equal
      // -ve if s comes before t
      // +ve if s comes after t
   String s1 = "abcdefg";
   String s2 = s1.substring(2,5);
      // s2 gets "cde"
   String s3 = s1.substring(3);
      // s3 gets "defg"
   int pos = s1.indexOf("c");
      // pos gets 2
   // String <-> int conversions
   int x = Integer.parseInt(s);
      // if s = "123", x gets 123
   String s = Integer.toString(x);
      // if x = 123, s gets "123"
```

**Simple Output:**
```
   System.out.println("x = " + x
     + " and y = " + y);
    //  x  and  y can be any type
```

**Simple Input with TextFile.KEYBOARD:**
```
   // fill this in
```

**Formatted Output with TextFile.SCREEN:**
```
   // fill this in
```

**File I/O with TextFile:**
```
   // fill this in
   // input files, output files
   // declaring variables, opening files
```

**Conditionals (Selection):**

**simple if:**
```
  if (a > b) {
    statement;
  }
```
**if … else:**
```
  if (a < b) {
    System.out.println("b is bigger");
    c = b;
  }
  else {
    System.out.println("a is bigger");
    c = a;
  } // end if
```
**if – else if – else if ....:**
```
   if (ch >= 'A' && ch <= 'Z') {
     System.out.println("upper case");
   }
   else if (ch >= 'a' && ch <= 'z') {
     System.out.println("lower case");
   }
   else if (ch >= '0' && ch <= '9') {
     System.out.println("digit");
   }
   else {
     System.out.println("other");
   } // end if
```

**while:**

```
   // computes 1 + 2 + ... + N
   int i = 0;
   while (i <= N) {
     sum += i; // same as sum = sum + i;
     i++;      // same as i = i + 1;
   } // end while
```

**for:**

```
   // same as preceeding while
   for (int i = 0; i <= N; i++) {
     sum += i;
   } // end for

   // prints odd numbers from 1 to 100,
   // in reverse order
   for (int i = 99; i > 0; i = i – 2)
        System.out.println(i);
```

**Math Class:**

```
// returns random double in [0,1)
Math.random();
// returns maximum of x and y
Math.max(x,y); // int & double versions
// returns minimum of x and y
Math.min(x,y); // int & double versions
```

**Class Structure (Application Program):**

```
public class MyProgram
{
  public static type mName(parameterList)
  {
  }
  public static void main(String args[])
  {
    ....
  }
}
```

**Class Structure (Object Class):**

```
public class SomeObject
{
  // instance variable
  private int property;
  // get (accessor) method
  public int getProperty(){
    return property;
  }
  // set (mutator) method
  public void setProperty(int val){
    property = val;
  }
  // constructor method
  public SomeObject(int val){
    property = val;
  }
  {
}
   // fill in use of SomeObject
   // in an application program
```

**Binary Search:**

```
   // fill this in
```

**Selection Sort:**

```
   // fill this in
```

**Recursion:**

```
   public static int factorial (int n){
     if (n == 0)
       return 1;
     else
       return n * factorial (n - 1); }
   // fill in other details, examples
```

**Number Types:**

```
   // integer: int is default
   // overflow & underfow wrap around
byte, short, int, long
   // floating point: double is default
float, double
   // mantissa bits limit signif digits
   // exponent bits limit range
   // adding, subtracting values very far
   // apart in magnitude risks loss of the
   // smaller when it is un-normalized so
   // digits of equal significance line up

   // fill in examples, other issues
```