Basic Java Cheat Sheet

For your reference; this sheet will also be included in exams

CISC 124, fall 2002

Variable Names:

```
studentName
middle_initial
student5
```

remember that upper & lower case matters!

Primitive Types:

```
int
double
char
boolean
```

Comparisons (primitive types only):

```
<   >     <=     >=     ==     !=
```

(Note comparison for equality is a double equal)

Strings:

```
String s = "hello, world"
String s2 = "abc" + "def" + 13;
   // s2 gets "abcdef13"
int len = s.length(); // len gets 12
char c = s.charAt(1); // c gets 'e'
if (s.equals(t))
   // true if t exactly the same as s
if (s.equalsIgnoreCase(t))
   // same as above but ignores case
int x = s.compareTo(t);
   // 0 if they're equal
   // positive if s is greater
   // negative if t is greater
String s1 = “abcdefg”;
String s2 = s1.substring(1,4);
   // s2 gets “bcd”
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”
```

Characters:

```
// if ch is lower case, capitalize
if (Character.isLowerCase(ch))
   ch = Character.toUpperCase(ch);
// returns true if ch is a decimal digit
// (i.e. ‘0’,’1’,...,’9’
if (Character.isDigit(ch))...
// translates ch into the equivalent int
// (0 to 9)
int i = Character.digit(ch, 10);
```

Arrays:

```
// create array of 10 doubles
double arr[] = new double[10];
number of elements in arr: arr.length
```

Increment/Decrement:

```
x++; // means x = x + 1;
x--; // means x = x – 1;
```

Extended Assignment:

```
x += 3; // means x = x + 3;
x -= 7; // means x = x – 7;
```

Declarations & Assignments:

```
int x;
x = 14;
double d = 15.2;
```

Output:

```
System.out.println("x = " + x
  + " and y = " + y);
```

// x & y can be any type

Input:

```
import hsa.*; // at start of class
String s = Stdin.readLine();
  // s gets entire line
String s = Stdin.readString();
  // s gets string up to white space
int i = Stdin.readInt();
double d = Stdin.readDouble();
char c = Stdin.readChar();
```

If Statements:

```
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);
```

Class Structure:

```
public class MyClass {
  public static void main(String args[]) {
    ....
  } // end main
} // end class MyClass
```

Math Class:

```
// returns random double in [0,1)
Math.random();
// returns the absolute value of x
Math.abs(x); // int & double versions
// returns square root of x
Math.sqrt(x)
// 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
```

Comparable:

```
public interface Comparable {
  // returns negative if this < o,
  //         0 if this = o,
  //         postiive if this > o
  public int compareTo(Object o);
}
```

Comparator:

```
public interface Comparator {
  // returns negative if o1 < o2,
  //         0 if o1 = o2,
  //         positive if o1 > o2
  public int compare(Object o1,
                     Object o2);
}
```

Vector:

```
constructors:
  Vector(int initialCapacity,
         int capacityIncrement)
  // capacityIncrement defaults to 0
  Vector(int initialCapacity)
  // initialCapacity defaults to 10
  Vector()
boolean add(Object o) // returns true
void add(int index,
                  Object element)
Object get(int index)
int indexOf(Object elem)
Object remove(int index)
Object set(int index, Object element)
int size()
```

Exceptions:

```
// Sample code that throws & catches
try {
  ... some code ...
  if (...some condition...)
    throw new Exception("optional msg");
  ... more code ...
}
catch (Exception e) {
  System.out.println("got an exception");
}
```

(last revised 20 Oct 2002)