UVa HSPC Java Cheatsheet

Primitive Data Types

`int`	32-bit signed two's complement integer
`long`	64-bit signed two's complement integer
`float`	32-bit floating point number
`double`	64-bit floating point number
`boolean`	Data type with two possible values: true or false
`char`	16-bit Unicode character

Operations

+	Arithmetic addition or String concatenation
-	Arithmetic subtraction
/	Arithmetic division
%	Integer division reminder (modulus)
++	Increment
--	Decrement
==	Equality
!=	Inequality
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
&&	Logical AND
!	Logical NOT
||	Logical OR

Variable Declaration and Assignment

```
int       index       =          0;
TYPE      NAME        ASSIGNMENT VALUE
```

If Statement

```
if ( Boolean Expression ){
      Statements;
}
```

While Loop

```
while ( Boolean Expression ){
      Statements;
}
```

For Loop

```
for ( Initialization ; Boolean Expression ; Increment){
      Statements;
}
```

Strings

```
String a = "UVa";
```
Creates the string a with value "Uva".

```
String b = "HSPC";
```
Creates the string b with value "HSPC".

```
boolean falseValue = a.equals(b);
```
a does not have the same value as b.

```
char letterU = a.charAt(0);
```
The first character of a is the letter 'U'.

```
int zero = a.indexOf("U");
```
The letter "U" is the first character in the string a.

```
int minusOne = a.indexOf("X");
```
The letter "X" does not appear in the string, returning -1.

```
String uvaHSPC = a + b;
```
The newly created string is "UVAHSPC".

Arrays

```
int[]   array   =   new   int[size];
ARRAY TYPE  NAME          ARRAY LENGTH
```

```
array[index] = 50;
int fifty = array[index];
```

Method Declaration

```
public  static   int   factoria (int n)
                                 l
VISIBILITY  CONTEXT  RETURN TYPE  METHOD NAME  ARGUMENTS
```

```
public static factorial(int n){
    /*body*/
}
```

UVa HSPC Java Cheatsheet

Math

All return doubles. Angles, unless otherwise specified are in radians.

`Math.E`	The base of the natural logarithm.
`Math.PI`	The ratio of the circumference of a circle to its diameter.
`Math.toDegrees(rad)`	Returns the angle rad in degrees.
`Math.toRadians(deg)`	Returns the angle deg in radians.
`Math.sin(ang)`	Computes the sine of ang.
`Math.cos(ang)`	Computes the cosine of ang.
`Math.tan(ang)`	Computes the tangent of ang.
`Math.asin(ang)`	Computes the inverse sine of ang.
`Math.log(a)`	The natural logarithm of a.
`Math.sqrt(a)`	The square-root of a.
`Math.pow(a,b)`	Raises a to the power of b.
`Math.round(a)`	Rounds a to the closest integer.
`Math.abs(a)`	Returns the absolute value a.
`Math.max(a,b)`	Returns the maximum of a and b.
`Math.min(a,b)`	Returns the minimum of a and b.

Scanner

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
```
Creates the a scanner object to ready from standard input (stdin).
```
int integer = scanner.nextInt();
```
Reads an integer from standard input.
```
String word = scanner.next();
```
Reads a string from standard input.
```
double number = scanner.nextDouble();
```
Reads a double from standard input.

Output

```
System.out.println("I'm printing! " +
dog);
```
Prints out a the string and the value of the variable dog with a new line.

Java Collections Framework

List

```
import java.util.*;
ArrayList<Integer> list = new
ArrayList<>();
```
Creates a new list of integers with an array-based implementation.
```
list.add(new Integer(1));
```
Adds the number 1 to the list.
```
System.out.println(list.get(0));
```
Prints the first element of the list, the number 1.

Set

```
import java.util.*;
HashSet<Integer> s = new HashSet<>();
```
Creates a set of integers.
```
s.add(new Integer(1));
```
Adds the number 1 to the set.
```
for(Integer i : s)
```
Iterates through each integer in the set.
```
System.out.println(i.toString());
```
Prints out each integer in the set.

Map

```
import java.util.*;
HashMap<String,String> k = new
HashMap<>();
```
Creates a mapping from strings to strings.
```
k.put("Dog","Cat");
```
Maps the string "Dog" (key) to "Cat" (value).
```
boolean true =
"Cat".equals(k.get("Dog"));
```
Retrieves the value for the key "Dog" and checks for equality with "Cat".