

## Java Extension Packages

```
import <class>
```

## Packages

```
package package_path.package_name;
```

## Common Extensions

```
java.awt, java.io, java.lang, java.util, javax.swing
```

## Data Types

```
boolean, char, byte, short, int, long, float, double, String
```

## Comments

```
// Single line Comment
```

```
/* Multiple line Comment */
```

## Arithmetic Operators

```
+ (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus)
```

## Equality Operators

```
== (Equal To),
```

```
!= (Not Equal To)
```

## Relational Operators

```
> (Greater Than), < (Less Than), >= (Greater Than or Equal To), <= (Less Than or Equal To)
```

## In-/Decremental Operators

```
++x (PreIncrement), x++ (PostIncrement), --x (PreDecrement), x-- (PostDecrement)
```

## Logical Operators

```
&& (logical AND), & (boolean logical AND), || (logical OR), | (boolean logical inclusive OR), ^ (boolean logical exclusive OR), ! (logical NOT)
```

## Escape Sequences

```
\n (newline)
```

```
\t (horizontal tab)
```

```
\r (carriage return)
```

```
\\ (backslash)
```

```
\" (double quote)
```

## Other

```
?: (Conditional)
```

```
= (Assignment)
```

## If Else

```
if (<condition>) {  
    <statement(s)>;  
}
```

```
else {  
    <statement(s)>;  
}
```

```
}
```

## Switch Case

```
switch(<expression>){
```

```
    case <option 1>:
```

```
        <statement>;
```

```
        break;
```

```
    case <option 2>:
```

```
        <statement>;
```

```
        break;
```

```
[default:
```

```
    <statement>;
```

```
]
```

```
}
```

## For Loop

```
for (<initial value>; <condition>; <in-/decrement>){  
    <statement(s)>;  
}
```

## While Loop

```
while (<condition> )  
{ <statement(s)>; }
```

## Do While Loop

```
do {  
    <statement(s)>;  
} while (<condition>);
```

## Arrays

```
int c[] = new int[5]; //declare and allocate in one
```

```
//declare and allocate in two
```

```
int myArray[ ];
```

```
myArray = new int[5];
```

```
//initialize
```

```
myArray = {10, 20, 30, 40, 50}
```

```
//access 3rd Element
```

```
myArray[2] = var;
```

## Method

```
<access modifier> <return data type> <function name> (<parameters>)
```

```
{
```

```
    <declarations>
```

```
    <statements>
```

```
    [return;]
```

```
    [return <expression>;]
```

```
}
```

## Class

```
<access modifier> <return data type> <class name> [extends
```

```
<superclass name>][implements <interface name>]
```

```
{
```

```
    <declarations>
```

```
    <methods>
```

```
}
```

## Exception Handling

```
try{
```

```
    //Code, can include method calls
```

```
}
```

```
catch(Exception e){
```

```
    //What to do on error. Multiple catches may be used
```

```
}
```

```
finally{
```

```
    //this code is executed with or without an error }
```

## File IO

```
// Read in a Text File
```

```
//should be contained in a try catch block
```

```
BufferedReader in = new BufferedReader(new
```

```
FileReader(directory.getPath()));
```

```
//directory is a File object
```

```
String nextLine = in.readLine(); //reads first line, repeat for next line
```

```
in.close();
```

```
// Write to a Text File
```

```
//should be contained in a try catch block
```

```
DataOutputStream out = new DataOutputStream(new
```

```
FileOutputStream(myfile.dat);
```

```
//creates myfile.dat, can add directory
```

```
out.writeUTF(theText); //writes String object theText
```

```
out.close();
```