HANDOUT 3.1

Reserved Words in Java

**Reserved Words Contextual Description**

**abstract** Used in a class definition to specify that a class is not to be instantiated, but rather only inherited by other classes

**boolean** Refers to an expression or variable that can have only a true or false value

**break** Terminates processing of a **switch** statement or loop

**byte** A sequence of eight bits

**case** Used in a **switch** statement to specify a match for the statement's expression

**catch** Used to specify the actions to be taken when an exception occurs (see **throw, try**)

**char** Declares objects whose values are characters

**class** Construct new types to describe data and operations

**continue** Used in a loop statement to transfer control to the beginning of the loop

**default** Used in a **switch** statement to handle expression values not specified using **case**

**do** Marks the beginning of a **do-while** statement

**double** Declares objects whose values are double precision real numbers

**else** Used as the alternative action of an **if** statement

**extends** Used to specify that a subclass inherits a superclass

**final** An entity that is defined once and cannot be changed or derived from later

**finally** Executes a block of statements regardless of whether a Java Exception, or run time error, occurred in a block defined previously by the "**try**" keyword

**float** Declares a primitive data type whose values are single precision real numbers

**for** Marks the beginning of a **for** statement

**if** Marks the beginning of an **if** statement

**implements** Optionally included in the class declaration to specify any interfaces that are used by the current class

**import** Used at the beginning of a source file that can specify classes or entire packages to be referred to later without including their package names in the reference

**instanceof** Tests whether the run-time type of its first argument is assignment compatible with its second argument

**int** Used to declare 32-bit integer numbers

**interface** Used to define a collection of method definitions and constant values

**long** Used to declare 64-bit integer numbers

**new** Allocates memory dynamically at run-time

**package** A group of classes of similar type

**private** Declares **class** members that are inaccessible from outside of the **class**

**protected** Declares **class** members that are accessible to derived classes and other classes in the same package

**public** Declares **class** members that can be accessed outside of the **class**

**return** Terminates a function, usually returning the value of some expression

**short** Used to declare 16-bit integer numbers

**static** Declares entities whose lifetime is the duration of the program

**super** Used to access members of a parent class

**switch** Marks the beginning of a switch statement

**synchronized** When applied to a method or code block, guarantees that at most one thread at a time executes that code

**this** Refers to the current object. Used within a **class** to unambiguously access other members of the **class**

**throw** Used to generate an exception (see **catch**, **try**)

**throws** Used in method declarations that specify which exceptions are not handled within the method but rather passed to the next higher level of the program

**try** Used to mark the beginning of a block containing exception handlers (see **catch**)

**void** Used to indicate the absence of any type (for a function or parameter list)

**while** Marks the beginning of a **while** statement, as well as the end of a **do-while** statement