# CS 1400 Exam 1 Guide

## Topics to Understand

How Java works (i.e. the Java Virtual Machine).

JVM analyzes and executes Java byte code

How Java compares to other languages.

The 8 primitive data types and know the sizes of the data, and the ranges of the data they can store.

* Boolean: 1 bit
* Byte: 8 bit ( 127 to -128 )
* Char: 16 bits
* Short: 16 bits ( 32,767 to -32,768 )
* Int: 32 bits ( 2,147,483,647 to -2,147,483,648 )
* Long: 64 bits ( 9,223,372,036,854,775,807 to -9,223,372,036,854,775,808 )
* Float: 32 bits ( 3.402,823,5 E+38 to 1.4 E-45 ) ie float hightemp, lowtemp; must use an f at the end or type cast it float b = (float) 12345.5
* Double: 64 bits ( 1.797,693,134,862,315,7 E+308 to 4.9 E-324 ) r = 10.8; // radius of circle   
  pi = 3.1416; // pi, approximately   
  a = pi \* r \* r; // compute area

The difference between a primitive data type variable/identifier and an object.

How to write data to the console.

System.out.print

How to read in data from the user on the console.

import java.util.Scanner;

Scanner stdIn = new Scanner(System.in);

stdIn.nextLine(), nextInt, stdIn.next().toLowerCase().charAt(0);

How to work with String objects, including comparison and manipulation.

String.format("%.1f", 1.234); String s1 = "Hello"

How to work with if, else if, and else.

Understand identifier scope.

**Identifiers** are the names of variables, methods, classes, packages and interfaces

How to work with switch statements.

How to work with and create primitive data types, reference data types, and constants

Three types of control structures – sequential, conditional, looping

Sequential: line-by-line execution

Conditional: if statements

Looping…

How to work with for loops, while loops, and do while loops.

How to decide which kind of loop should be chosen for implementation.

How to work with methods.

Be able to read code on paper and understand what the logical flow of the program will be if it were to be run on a computer.

Be able to write code by hand on paper so that if it were placed on a computer, it would compile and run.

public static void main(String args[])

How to work with the Random number generator class.

**import** java.util.Random;

Random random = **new** Random();

random.nextInt(100);

rn.nextInt(10) + 1; // shows random 1 to 10, if you want a negative, subtract the lowest random number you might get with the lowest number you want.

uses the math random

math random 0 to .9

How to work with the Scanner class for gathering user input

Scanner stdIn = new Scanner(System.in);

How to work with the Modulus operator

% remainder

How to validate user input

* [hasNext()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNext%28%29) - does it have *any* token at all?
* [hasNextLine()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNextLine%28%29) - does it have another line of input?
* For Java primitives
  + [hasNextInt()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNextInt%28%29) - does it have a token that can be parsed into an int?
  + Also available are [hasNextDouble()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNextDouble%28%29), [hasNextFloat()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNextFloat%28%29), [hasNextByte()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextByte%28%29), [hasNextShort()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextShort%28%29), [hasNextLong()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextLong%28%29), and [hasNextBoolean()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextBoolean%28%29)
  + As bonus, there's also [hasNextBigInteger()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextBigInteger%28%29) and [hasNextBigDecimal()](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html" \l "hasNextBigDecimal%28%29)
  + The integral types also has overloads to specify radix (for e.g. hexadecimal)
* Regular expression-based
  + [hasNext(String pattern)](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNext%28java.lang.String%29)
  + [hasNext(Pattern pattern)](http://java.sun.com/javase/6/docs/api/java/util/Scanner.html#hasNext%28java.lang.String%29) is the [Pattern.compile](http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html" \l "compile%28java.lang.String%29) overload

How to declare and work with constants

public static final int MAX\_SECONDS = 25;

Standard naming conventions for class names, method names, variable names

Class: start with upper

Method: start with lower and be a verb, actionPerformed(), print()

Variable: start with a lower

Numeric operators and order of precedence

How to work with increment and decrement operators

How to perform numeric type conversions with explicit casting

char/short/int/long/byte

float/double

boolean

How to compile source code into bytecode using the javac command

javac HelloWorld.java

## Vocabulary Terms to Know

Algorithm: a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Application Program Interface (API): a set of [subroutine](https://en.wikipedia.org/wiki/Subroutine) definitions, [protocols](https://en.wikipedia.org/wiki/Protocol), and tools for building [application software](https://en.wikipedia.org/wiki/Application_software).

Assignment statement: sets and/or re-sets the value stored in the storage location(s) denoted by a variable name

Assignment operator: assigns the value of its right operand to its left operand

Cast operator ()

Byte code:  computer object **code** that is processed by a program, usually referred to as a virtual machine, rather than by the "real" computer machine, the hardware processor.

Comments (in-line and block)

Compiler

Integrated Development Environment (IDE)

Java Development Kit (JDK): **JDK** is a superset of **JRE**, and contains everything that is in **JRE**, plus tools such as the compilers and debuggers necessary for developing applets and applications.

Java Runtime Environment (JRE): **JRE** is the container, **JVM** is the content. **Java Runtime Environment** contains **JVM**, class libraries, and other supporting files. It does not contain any development tools such as compiler, debugger, etc. **JRE** = **JVM** + Java Packages Classes (like util, math, lang, awt, swing etc) + runtime libraries

Java Virtual Machine (JVM)

Programming language

Source code

Object code: code produced by a compiler or assembler.

Pseudocode

Identifiers: the names of variables, methods, classes, packages and interfaces, In the HelloWorld program, HelloWorld , String , args , main and println are **identifiers**

Variable

Operator

Prompts

Sequential Statements

Conditional statements

Tracing

Prologue

Named constant:  A **named constant** is exactly like a variable, except that its value is set at compile time (by initializing it) and CANNOT change at runtime

Final modifier

Modulus operator