**Ch. 1 - Ch.3 Definitions**

**Short Notes:**

**Application Program Interface ( API)**

A library in Java that contains predefined classes and interfaces

**Assembler**

a device used to translate assembly-language programs into machine code

**Assembly Language**

uses a short descriptive word to represent each of the machine-language instructions

**Bit**

Binary digits

**Block**

anything inside of a {xxxxxx}

**Block Comment**

/\***XXXXXXXXXXXXX**\*/

**Byte**

8 bits to 1 byte

**Bytecode**

Similar to machine instructions, but can run on any platform with a JVM

**bytecode verifier**

checks the validity of a bytecode

**comment**

on a program denoted by //xxxxx or /\***xxxx**\*/

**compiler**

translates a Java source file into a Java bytecode file

**dot pitch**

the amount of space between pixels, measured in millimeters

**Integrated Development Environment (IDE)**

An environment for developing Java programs

**Interpreter**

translates source code into machine code

**Java Development Toolkit**

consists of a set of separate programs, each invoked from a command line, for developing and testing Java programs

**keyword**

Reserved words that have a specific meaning in java and cannot be used for variables

**logic error**

occurs when a program does not perform the way it was intended to

**runtime error**

errors that cause a program to terminate early, an impossible operation is detected

**source code/program**

a high-level program's code

**statement**

instructions for a high-level program

**statement terminator**

;

**syntax error**

result from errors in code construction, such as misspellings, wrong punctuation, etc.

**input error**

Occurs when the user inputs a value the program cannot handle

**assignment operator**

=

**assignment statement**

evaluates to the value to be assigned to a variable (=)

**byte type**

-128 to 127

**casting**

an operation that converts a value of one data type into a value of another data type

**constant**

a number in the program that never changes, denoted by "final"

**data type**

the kind of data stored in each variable

**decrement operator**

--

**double type**

real numbers, decimal places, twice as precise as float

**expression**

represents a computation involving values, variables, and operators that, taking them together, evaluates to a value

**final keyword**

denotes a value as a constant

**floating-point number**

Numbers with a decimal point (var double)

**identifier**

-Names that refer to values or names - letters, digits, \_, and $.

**Increment operator**

++

**int type**

an exact number, 1, 4 or 10

**IPO**

input, process, output - describes simple code

**literal**

a constant value that appears directly in a program

**long type**

a very large int, more precise

**narrowing (of types)**

Casting a data type from a large range to a smaller range - Java will automatically widen a type, but we must narrow a type explicitly.

**Operands**

The values operated on by an operator

**operators**

+, -, \*, /, %

**overflow**

If you try to store a value in a data type that cannot handle it.

**postdecrement**

-- placed after variable. uses original variable in expression then decreases by 1

**postincrement**

++ placed after variable. uses original variable in expression then increases by 1

**predecrement**

-- placed before variable. decreases variable by one, then uses it in the expression

**preincrement**

++ placed before variable. increases variable by one, then uses it in the expression

**primitive data type**

‘int’, real numbers, characters and booleans

**Scope of a variable**

The part of a program where the variable can be referenced

**variable**

represents a value stored in the computer’s memory

**widening (of types)**

casting from a small type to a larger type, computer does this automatically.

**wildcard import**

imports all the classes in a package by using a \* **(import java.util.**\*;)

**concatenate strings**

using the (+) sign to combine strings

**Boolean Expression**

An expression that evaluates a Boolean value to be true or false

**Boolean Value**

can be true or false

**conditional operator**

? : for if statement shorthand (ternary operator)

**dangling else ambiguity**

when else matches with the most recent if statement

**fall-through behavior**

using no breaks in a switch

**operator associativity**

determines the order in which operators are evaluated

**operator precedence**

var++, + and -, casting, !, \* / %, + - concatenation, (See page 105)

**selection statement**

statements that let you choose actions with alternative choices

**short circuit operator**

same as lazy operator - && or || (and, or)

**boolean operators**

!, &&, ||, ^

**Augmented assignment operators**

+=, -=, \*=, /= and %= (i+= 8 is i = i + 8)

**Conditional Expression ( ? :)**

Evaluates an expression based on a condition (pg. 103)

**Relational Operators (Boolean)**

<, <=, ==, !=, >, >=

**increment operator**

++

**keyword**

abstract is a

**escape character**

\n

**preprocessor**

import statement

**directive**

import

**identifier**

Variable

**name of type**

int

**Literal**

Constant value directly in a program that stands for itself

**keywords**

do, else, and break

**name of a type**

float

**illegal identifier**

4thQtrSales

**octa integer**

075

**name of type**

char

**escape sequence**

**\"**

**Floating point**

scientific notation

**String**

a type

**floating point/pi**

3.14159E1

**variable name**

bool

**util**

a class name in the system library that contains different java functions

**nextDouble**

method that is applied to objects of Scanner

**final**

keyword that are in front of a constant. We use ‘final’ keyword to make a variable to a constant

Short Questions:

1. Similar to machine instructions, but can run on any platform with a JVM

###### Ans. Bytecode

1. represents a value stored in the computer’s memory

Ans. variable

1. instructions for a high-level program

Ans. Statement

1. Numbers with a decimal point (var double)

Ans. floating-point number

1. Int

Ans.name of type

**MCQ:**

1. using no breaks in a switch
   1. conditional operator
   2. **fall-through behavior**
   3. short circuit operator
   4. floating-point number
2. float
   1. **name of a type**
   2. Escape character
   3. data types
   4. cope of a variable
3. =
   1. widening (of types)
   2. concatenate strings
   3. **assignment operator**
   4. conditional operator
4. on a program denoted by //xxxxx or /\***xxxx**\*/
   1. compiler
   2. **comment**
   3. int type
   4. variable
5. Similar to machine instructions, but can run on any platform with a JVM
   1. **Bytecode**
   2. Assembler
   3. Literal constant
   4. Source code
6. Reserved words that have a specific meaning in java and cannot be used for variables
   1. **Keyword**
   2. predecrement
   3. Literal
   4. final keyword
7. consists of a set of separate programs, each invoked from a command line, for developing and testing Java programs
   1. dangling else ambiguity
   2. narrowing (of types)
   3. **Java Development Toolkit**
   4. assignment statement
8. an operation that converts a value of one data type into a value of another data type
   1. identifier
   2. Assembler
   3. **casting**
   4. overflow
9. +, -, \*, /, %
   1. Operands
   2. **operators**
   3. Preprocessor
   4. keyword
10. A library in Java that contains predefined classes and interfaces

**Application Program Interface (API)**

1. represents a computation involving values, variables, and operators that, taking them together, evaluates to a value
   1. **expression**
   2. identifier
   3. Assembler
   4. predecrement
2. \n
   1. increment operator
   2. name of type
   3. variable name
   4. **escape character**
3. int, real numbers, characters and booleans
   1. **primitive data type**
   2. widening (of types)
   3. selection statement
   4. operator precedence
4. 8 bits to 1 byte
   1. **Byte**
   2. Bytecode
   3. Machine code
   4. int
5. the values operated on by an operator
   1. **operands**
   2. operators
   3. primitives
   4. literal
6. !, &&, ||, ^
   1. **Boolean operators**
   2. preprocessor
   3. Boolean Value
   4. increment operator
7. when else matches with the most recent if statement

**dangling else ambiguity**

1. ++ placed after variable. uses original variable in expression then increases by 1

**postincrement**

1. 3.14159E1

**floating point/pi**

1. represents a value stored in the computer’s memory an exact number, 1, 4 or 10

**int**

1. a number in the program that never changes, denoted by "final"
   1. variable
   2. **constant**
   3. int
   4. data type
2. the amount of space between pixels, measured in millimeters
   1. dotted
   2. **dot pitch**
   3. byte code
   4. byte
3. -128 to 127
   1. **Byte type**
   2. Int type
   3. Long type
   4. Short type