This guide contains a list of topics that may be tested on the exam, general conceptual questions for each topic, and some more concrete questions. I recommend looking through the topics and, for any topic you’re not fully comfortable with, checking your understanding with conceptual questions. I also recommend that you complete a variety of the specific questions.

You should be prepared to understand the execution of, correct, and write code for this exam.

**Topic Outline:**

- **Hardware components and their interactions**
  - CPU
  - Motherboard
  - RAM
  - Hard drive

- **Compiling and executing: the compiler and JVM**
  - The commands javac, java
  - Machine code, bytecode, and Java code
  - File name requirements

- **Syntax basics**
  - Semicolons ; and curly braces {}
  - Keywords
  - Identifier requirements
  - Comments: block and inline

- **Data types**
  - Variables
  - Primitive types: int, double (including # of bits); long, short, and byte; char; boolean
  - Reference type: String
  - Casting

- **Numeric Operators**
  - +, -, /, *, % on various data types
  - Overflow
  - Char as a “numeric-like” type.

- **Assignment, increment, and decrement operators**
  - =, /=, *=, +=, -=, %=
  - Pre- and post- increment and decrement, as in x++, ++x, --x, x--
• String operators escape characters
  – Concatenation +
  – \n, \, \', and "

• Scanners
  – Importing
  – Scanner initialization
  – Scanner methods for reading input

• Booleans and their operators
  – Boolean literal (note case)
  – Comparison operators: >, <, >=, <=, ==, !=
  – Logical operators: !, &&, ||, ^

• If, Else, and Else-If
  – Single and block forms
  – General syntax and use

• Switch Statements
  – General syntax and use
  – Types permitted for the switch expression
  – Use of break;

• While loops (do-while and normal)
  – Syntax and use
  – Steps taken when a while loop is reached
  – Understanding iterations in while loops
  – Infinite loops

• For Loops
  – The statements in the start of a for loop
  – Steps taken when a for loop is reached
  – Single versus block for loops
  – Scope of variables in for loops

General Conceptual Questions:

• Hardware components and their interactions:
  1. What is the purpose of each of the CPU, motherboard, RAM, and hard drive?
  2. How is each component involved in the execution of a program?
  3. What are the relative capacities and speed of the CPU registers, RAM, and hard drive?
4. What are the ALU and CU? What do they do?

• Compiling and executing

1. What do the commands javac and java do?
2. What are the compiler and JVM? How do they interact, if at all?
3. What is the difference between the physical machine’s code, Java bytecode, and Java language code?
4. How do you determine what name a Java code file needs based on the content of the program?

• Syntax

1. What do curly braces { } signify?
2. What are semicolons ; used for?
3. What Java keywords have we discussed so far? Note capitalization.
4. What makes a valid identifier? What conventions are there surrounding symbols and capitalization of identifiers?
5. How do you make an inline comment? A block (multi-line) comment?

• Data Types

1. What is the difference between a variable and a data type?
2. What does a variable declaration do?
3. What are the size, in bits, of int and double?
4. What cross-type assignments (ex. double b = 5;) are allowed? What require casting, or cannot even be cast?
5. What types do various literal values have?
6. What is the syntax for casting?
7. Which of the numeric (and numeric-like) operators can be converted safely without casting? Which need casting?
8. Give examples of invalid String and character literals.

• Numeric Operators

1. What is the order of operation for various data types?
2. What is the result of mixing (or not mixing) int and double types for various operators? What about long, byte, short, and float?
3. What is the rule for integer division?
4. What is overflow?
5. What numeric operations can be performed on chars?
• Assignment, Increment, and Decrement
  1. How are the various assignment operators used?
  2. How can the increment and decrement operators be used in numeric (or conditional) expressions?
  3. What is the intermediate value for pre- and post- increment and decrement operators?
  4. What happens to the value of a variable when a pre- or post-increment or decrement operator is used in a numeric expression or conditional expression?

• String Methods and Escape Characters
  1. How do you concatenate two Strings?
  2. What is the precedence of the concatenation operation compared to numeric operations?
  3. What does the escape character \n do? What happens if you print a string with a \n in the middle of it?
  4. When would you want to use \’ or "? How do you use them in a character or String literal?

• Scanners
  1. How do you import the Scanner library?
  2. How do you declare and initialize a Scanner variable?
  3. What do some of the Scanner methods for reading input do? What is the difference between next() and nextLine()?
  4. Given Scanner input, how do you use Scanner methods to read input from a user?

• Booleans and Boolean Operators
  1. What are the two boolean literals? (Be careful of capitalization errors.)
  2. What do each of the comparison operators (> , < , >= , <= , == , !=) do? How do you use them, and what type results from using them?
  3. What is the difference between = and ==?
  4. What do each of the logical operators (! , && , || , ^) do? What are the results of various combinations of boolean inputs to each operator?
  5. How can you combine comparison and logical operators?

• If, Else, and Else-If
  1. How can if and else statements be combined? Are there any restrictions on ordering?
  2. What is the general syntax for block and single-statement if statements?
  3. What data types can the conditional expression of an if statement have?
• Switch Statements
  1. What is the general syntax of a switch statement? Be careful about symbols like (), {}, and :
  2. What data types are permitted in the switch expression?
  3. What happens if a break statement is omitted from a case in a switch statement?
  4. When does the default case of a switch statement get executed?
  5. Why would you want to use a switch statement?
• While Loops
  1. What is the general syntax of a while loop?
  2. What happens, step-by-step, when a while loop is reached in code? When does the loop terminate?
  3. How can you use while loops to iterate a fixed number of times?
  4. How can you use while loops to iterate based on user input?
  5. How can you make an infinite loop?
• For Loops
  1. What are the three statements that are in within the parentheses () of a for loop’s start? What does each do, and when are they checked/used?
  2. What is the difference versus a single statement and a block for loop? Why should you typically use a block for loop?
  3. If you declare a variable within a for loop, is it usable in the three statements that start a for loop? How about after the loop?
  4. If a variable is declared in a for loop’s initial action or action after iteration, what scope does it have?
  5. How can you make an infinite loop with a for loop?

Specific Practice Questions:

These questions are intended to be worked out on paper, but I would also recommend checking your answer by writing, compiling, and running Java code.

1. (Syntax) What is wrong with the following code statement? Correct it so that it compiles as a single statement in a Java program.

   int x == 45.

2. (Loops) Write a loop that prints the numbers 0 to 100.
3. (If statements, conditional operators) What does the following code snippet print?

```java
boolean bool1 = true;
boolean bool2 = false;
int x = 2;

if(x < 2){
    if(bool1 || (bool2 ^ bool2)){
        System.out.println("Beep.");
    } else if(x+5 > 6 && bool1){
        System.out.println("Bop.");
    }
} else {
    if((bool1 && bool2) ^ (bool1 || bool2)){
        System.out.println("Meep.");
    } else {
        System.out.println("Mop.");
    }
}
```

4. (Switch statements, Scanners, if-else) Write a code snippet that takes user input for an integer and, using a switch statement, prints -2 if the user enters -1, "Wow!" if the user enters 10, and prints 0 otherwise.

Repeat this problem, using if-else statements instead of a switch statement.

5. (Numeric operations) What do the following expressions evaluate to? Assume the following variable declarations/values:

```
int x = 10, y = 2, z = 3; // (Note: this is valid syntax.)
double a = 2.5, b = 3.3;
String greet = "Hello!";
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x/y</td>
<td></td>
</tr>
<tr>
<td>z + b * y</td>
<td></td>
</tr>
<tr>
<td>x &gt; b * z &amp;&amp; a + b &gt; y * z</td>
<td></td>
</tr>
<tr>
<td>(int) a * b</td>
<td></td>
</tr>
<tr>
<td>a++</td>
<td></td>
</tr>
<tr>
<td>++y != z</td>
<td></td>
</tr>
<tr>
<td>true ^ (y &gt; (int) a)</td>
<td></td>
</tr>
</tbody>
</table>
6. (Scanners, loops, if/else, calculations) Write a program that prompts users for a positive integer number. The program should print out the last \( N \) digits of the number, where \( N \) is the first digit of the number input. The number entered is guaranteed to be a positive value in the range of \texttt{int}. A sample run is shown below.

Enter a number: 5894230
94230