Exam 2 – Study guide

The following are suggested study topics for the multiple-choice section of the test.

- Understand the concept of conditionals in Java, which are used to make decisions in code based on conditions.
- If Statements:
    - Learn how to use if statements to conditionally execute a block of code.
- Understand the syntax of if statements, including the use of conditions in parentheses.
- If-Else Statements:
    - Study if-else statements, which allow you to execute different code blocks based on a condition.
- Learn the syntax and how to handle multiple conditions with nested if-else statements.
- Learn about the switch statement, used for multi-way branching based on a specific value.
- Understand the syntax and how to handle different cases in switch case.
- Ternary Operator:
    - Study the ternary conditional operator (? :), which provides a compact way to write simple conditional expressions.
- Logical Operators:
    - Explore logical operators such as && (logical AND), || (logical OR), and ! (logical NOT).

- Understand how to combine conditions using logical operators.
- Comparison Operators:
    - Learn about comparison operators like ==, !=, <, >, <=, and >=.
- Understand how to compare values and use them in conditionals.
- Understand how to create complex conditions by combining multiple conditions using logical operators.
- Be aware of common mistakes when using conditionals, such as assignment (=) vs. equality (==) operators.
- Introduction to Loops:
    - Understand the concept of loops in Java, which are used for repetitive execution of code blocks.
- While Loops:
    - Learn how to use while loops to repeat a block of code as long as a specified condition is true.
    - Understand the syntax of while loops, including initialization, condition, and update.
- Do-While Loops:
    - Study do-while loops, which guarantee that the block of code will execute at least once, as the condition is checked after execution.
    - Learn the syntax of do-while loops and when to use them.
- For Loops:
    - Explore for loops, which are used for iterating a specific number of times.
    - Understand the syntax of for loops, including initialization, condition, and update.

- Nested Loops:
    - Study the concept of nested loops, where one loop is placed inside another.
- Loop Termination:
    - Understand how to ensure that loops terminate by properly defining exit conditions.
- Be aware of infinite loop pitfalls.
- Introduction to Classes:
    - Understand the fundamental concept of classes in Java, which are the building blocks of object-oriented programming.
- Class Declaration:
    - Learn how to declare a class in Java, including the class name, access modifiers, and the class keyword.
- Class Members:
    - Study class members, including fields (attributes) and methods (functions).
    - Understand how to declare and access class members.
- Constructors:
    - Explore constructors, which are special methods used to initialize objects.
    - Learn how to create constructors with different parameters and default constructors.
- Method Overloading:
    - Understand method overloading, which allows multiple methods with the same name but different parameter lists in a class.

- Access Modifiers:
- Study access modifiers (public and private) and their impact on class members' visibility and access.
- Encapsulation:
    - Understand the concept of encapsulation, which involves bundling data (fields) and methods into a single unit (a class).
    - Learn how to use accessors (getters) and mutators (setters) to control access to class fields.
- Static Members:
    - Study static members (fields and methods), which are associated with the class rather than specific instances.
    - Understand the use of the static keyword.
- Understand the importance of toString method and how to write and use them in a program.
- Dependency and Aggregation:
    - Explore the concepts of dependency and aggregation, which involve creating relationships between classes.
- Basic concept and use of UML diagrams

The following are suggested study topics for the programming section of the test:

1. Learn how to write a class including the following members:
    - Variables (attributes)
    - getters and setters methods
    - constructors
    - toString method
2. How to design and implement methods to solve a problem
3. Create objects of the class
4. Call (invoke) different methods of a class
5. Enforce encapsulation
6. How to design a while loop using sentinel value
7. You may have to use if/else statements as well