CODE ANALYSIS C

1. **DESCRIPTION:** Given a set of computer programs or parts of programs, teams will determine the output that each would produce.

   TEAM OF UP TO: 2 **EYE PROTECTION:** No **IMPOUND:** No **APPROX TIME:** 50 minutes

2. **EVENT PARAMETERS:**
   - Satellite/Traditional format:
     i. Each team is permitted one 8.5”x11” sheet of paper, which may be in a sheet protector sealed by tape or laminated, that may contain information on both sides in any form and from any source without any annotations or labels affixed.
     ii. Competitors may NOT bring electronic devices with keyboards, keypads, or touchscreens into the testing room, including computers, calculators, and cell phones.
   - miniSO format: Each teammate is allowed one 8.5”x11” sheet of paper. Competitors are not allowed to use any other electronic device other than to view the exam and record their answers.
   - All computer code shall be given in the Java computer language. Problems should focus on general programming principles, rather than the peculiarities of the Java language.
   - Code may be provided as a series of statements to be executed in the order given, as a method or methods to be called with a given set of arguments, or as a complete program with a main() method.
   - All input or data values shall be specified in the given code, or as a given set of arguments to a method call. No other form of input is permitted, including command-line arguments, System.in, external files, or graphical user interfaces.
   - Competitors shall determine the output of the code, which will either be the return value of a method call, or the text created by System.out.print(), System.out.println(), and/or System.out.printf() statements. Space shall be provided for competitors to write their results. If a problem's score depends on the number or spacing of output lines, then the answer space shall include horizontal writing lines. If a problem is also scored by the number or spacing of characters, then the answer space shall include a grid. If more answer space is provided than is needed, competitors must leave the extra space blank.

3. **PERMITTED TOPICS:**
   - **i**ntegers, unary operators + and -, and binary operators + - * / %.
   - characters. Character literals must be in the range of ' ' (space) to '~' (tilde), plus escape sequences '\0' (null), '\n' (newline), '\\' (backslash), '\'' (single quote), and '\"' (double quote). Addition, subtraction, and comparison of characters are permitted as long as the result is in one of the ranges '0' to '9', 'A' to 'Z', or 'a' to 'z'. Competitors shall not be expected to know ASCII or other character codes.
   - boolean, true, false, unary operator !, binary operators == != < <= > >= && || ^, and tertiary operator ?:.
   - Arrays of fixed size, initializer lists, setting and getting elements using [], and the method length.
   - Strings; string literals made from the permitted character literals; methods length, equals, equalsIgnoreCase; concatenation with +.
   - Variable declarations and definitions. No more than 10% of the score may involve issues of variable scope.
   - The assignment operators (for example, +=), pre- or post-increment (++), and pre- or post-decrement (--) may be used, but may NOT involve side-effects (for example, x++ − x).
   - Operator precedence and associativity. (*type*)-casting.
   - {} blocks, if, if...else, while, do...while. Code shall be properly-indented (except for up to 10% of the score). // and /* */ style comments may be used.
   - Method definitions, arguments to a method, return values, recursion.
   - System.out.printf() is limited to fixed text and the format specifiers %%, %c, %d, and %s.

- Topics reserved for the state and national tournament level are for loops (loop variables may not be modified inside the loop); and the System.out.printf() format modifiers *width.precision*, +, -, and 0.
- Topics reserved for the national tournament are the bitwise operators ~ & | ^, binary constants 0n*nnn*, hexadecimal constants 0x*nnn*, break, continue, and switch blocks (code must not fall through to another case).
- For this year, problems may also include algorithms on searching and sorting arrays.

4. FORBIDDEN TOPICS:

- import statements, class definitions, and static may be included to make the code a complete program, but competitors are not expected to interpret their significance.
- assert, try...catch.
- Classes, inheritance, polymorphism, new, constructors, method overloading, instance variables, this, finalize.
- Methods of the Array, Arrays, and ArrayList classes.
- enums, static, variable-length argument lists, regular expressions, Collections, Lists, Sets, resizing, or dynamic memory management.
- File and network handling, multithreading, web programming.
- Any feature of Java that is non-standard, or which varies among versions or platforms.

5. SCORING:

- Each problem shall specify its number of points. Partial credit where appropriate is strongly recommended.
- The first problem shall be worth 20% to 40% of the total score. Competitors will turn their answer in when finished, and the time shall be recorded.
- Remaining problems shall be worth no more 20% of the total score each.
- Ties shall be broken first by the highest number of points on the first problem, second by the least amount of time on the first problem, and finally by the highest score on designated problems.

6. RECOMMENDED RESOURCES:

- Gaddis, Tony. *Starting Out With Java*. 4th ed. Addison-Wesley, 2011. ISBN 978-0-132-16476- 4.
- Lewis, John et al. *Java Software Solutions for AP Computer Science*. 3rd ed. Addison-Wesley, 2011. ISBN 978-0-13-137469-0.
- Sedgewick, Robert and Kevin Wayne. *Algorithms*. 4th ed. Addison-Wesley, 2011. ISBN 978-0-321-57351-3. <http://algs4.cs.princeton.edu/code>
- The International Obfuscated C Code Contest <http://www.ioccc.org> gives plenty of examples of programs that are deliberately made difficult to analyze.