IT Skill Test GIC Myanmar

Duration: 30 Minutes *Total Questions: 8*

---

1. You are developing a user authentication system. The requirement is to lock a user's account after three failed login attempts. Which conditional logic would be most appropriate to implement this feature?

A. if (failedAttempts == 3) { lockAccount(); }

B. while (failedAttempts < 3) { allowLogin(); }

C. switch (failedAttempts) { case 3: lockAccount(); break; }

D. if (failedAttempts >= 3) { lockAccount(); }

2. In a temperature monitoring system for a industrial freezer, you need to trigger an alert if the temperature rises above -5°C or falls below -20°C. Which conditional statement would be most appropriate?

A. if (temperature > -5 || temperature < -20) { triggerAlert(); }

B. if (temperature > -5 && temperature < -20) { triggerAlert(); }

C. if (temperature <= -5 || temperature >= -20) { triggerAlert(); }

D. switch (temperature) { case > -5: case < -20: triggerAlert(); break; }

3. You're implementing a simple state machine for a traffic light. The light should change from Green to Yellow, Yellow to Red, and Red back to Green. Which control structure would be most appropriate for this cyclic behavior?

A. if (currentState == GREEN) { nextState = YELLOW; } else if (currentState == YELLOW) { nextState = RED; } else { nextState = GREEN; }

B. while (true) { if (currentState == GREEN) nextState = YELLOW; else if (currentState == YELLOW) nextState = RED; else nextState = GREEN; }

C. switch (currentState) { case GREEN: nextState = YELLOW; break; case YELLOW: nextState = RED; break; case RED: nextState = GREEN; break; }

D. for (int i = 0; i < 3; i++) { if (i == 0) nextState = YELLOW; else if (i == 1) nextState = RED; else nextState = GREEN; }

4. In a file processing application, you need to read lines from a file until you encounter an empty line or reach the end of the file. Which loop structure would be most suitable for this task?

A. for (String line = reader.readLine(); line != null; line = reader.readLine()) { processLine(line); }

B. while ((line = reader.readLine()) != null && !line.isEmpty()) { processLine(line); }

C. do { line = reader.readLine(); processLine(line); } while (line != null);

D. for (String line : reader.readLines()) { if (line.isEmpty()) break; processLine(line); }

5. You're developing a grading system that assigns letter grades based on numerical scores. The grading scale is: A (90-100), B (80-89), C (70-79), D (60-69), F (0-59). Which conditional structure would be most appropriate for this task?

A. if (score >= 90) grade = 'A'; else if (score >= 80) grade = 'B'; else if (score >= 70) grade = 'C'; else if (score >= 60) grade = 'D'; else grade = 'F';

B. switch (score / 10) { case 10: case 9: grade = 'A'; break; case 8: grade = 'B'; break; case 7: grade = 'C'; break; case 6: grade = 'D'; break; default: grade = 'F'; }

C. grade = (score >= 90) ? 'A' : (score >= 80) ? 'B' : (score >= 70) ? 'C' : (score >= 60) ? 'D' : 'F';

D. while (score >= 60) { if (score >= 90) { grade = 'A'; break; } if (score >= 80) { grade = 'B'; break; } if (score >= 70) { grade = 'C'; break; } grade = 'D'; break; } if (grade == null) grade = 'F';

6. In a web application, you need to validate user input for a username. The username must be between 3 and 20 characters long and contain only letters, numbers, and underscores. Which conditional statement would best implement this validation?

A. if (username.length() >= 3 && username.length() <= 20 && username.matches("^[a-zA-Z0-9_]+$")) { /* valid */ } else { /* invalid */ }

B. if (username.length() > 2 || username.length() < 21 && username.matches("\w+")) { /* valid */ } else { /* invalid */ }

C. switch (username.length()) { case 3: case 4: /* ... */ case 20: if (username.matches("^[a-zA-Z0-9_]+$")) { /* valid */ } break; default: /* invalid */ }

D. while (username.length() >= 3 && username.length() <= 20) { if (username.matches("^[a-zA-Z0-9_]+$")) { /* valid */ break; } else { /* invalid */ break; } }

7. You're developing a method to calculate the factorial of a number. Which implementation would be most efficient and appropriate for numbers up to 20?

A. public static long factorial(int n) { if (n <= 1) return 1; else return n * factorial(n - 1); }

B. public static long factorial(int n) { long result = 1; for (int i = 2; i <= n; i++) result *= i; return result; }

C. public static long factorial(int n) { return IntStream.rangeClosed(1, n).reduce(1, (a, b) -> a * b); }

D. public static long factorial(int n) { return LongStream.rangeClosed(1, n).reduce(1, (a, b) -> a * b); }

8. You are developing a Java application for a library management system. The system needs to categorize books based on their age and condition. Books older than 10 years are considered 'Vintage', books between 5 and 10 years old are 'Standard', and books less than 5 years old are 'New'. Additionally, if a book's condition is 'Poor', it should be marked for 'Restoration' regardless of its age. Which conditional logic structure would be most appropriate for this categorization?

A. if (bookAge > 10) { category = "Vintage"; } else if (bookAge > 5) { category = "Standard"; } else { category = "New"; }

B. switch (bookAge) { case > 10: category = "Vintage"; break; case > 5: category = "Standard"; break; default: category = "New"; }

C. if (condition.equals("Poor")) { category = "Restoration"; } else if (bookAge > 10) { category = "Vintage"; } else if (bookAge > 5) { category = "Standard"; } else { category = "New"; }

D. while (bookAge > 0) { if (bookAge > 10) { category = "Vintage"; } else if (bookAge > 5) { category = "Standard"; } else { category = "New"; } break; }