Code inspection

1. Check comparison between objects (no == but equals)

All compared right manner ,properly usage of == only when an object may not be instantiated

1. Output displayed free of spelling and grammatical errors

No output in method assigned

1. Error messages comprehensive and guidance how to correct them
2. Output well formatted in terms of line stepping and spacing

No output in methods assigned

1. No brutish programming(link: http://users.csc.calpoly.edu/~jdalbey/SWE/CodeSmells/bonehead.html )
   * 1. No wrong power,
     2. no wrong letter conversion,
     3. no wrong digit recognition,
     4. appropriate usage of Boolean condition,
     5. return Boolean value based on Boolean condition?( **check is locked method and copy resource**
     6. initialize an array. No array
     7. Assign a value to a variable based on the value of a second variable. None of this kind of action applied
     8. Convert a string representation of a number into a Java int. no conversion of a string into an int
     9. Advance the turn among players in a multiplayer game (or any situation that needs to maintain a counter that "rolls over"). No turn of something of this kind
     10. Create a formatted phone number from the string of digits. No phone number
     11. Maintain an ArrayList in sorted order
     12. Performing an algebraic transformation on some value
     13. Printing the time in MM:SS format. No time
     14. Removing certain characters from a string.   
         A common task is to replace or remove certain characters from a string, e.g, remove vowels
     15. Use named constants instead of numeric literals.   
         Many coding standards recommend avoiding numeric constants (literals) and replace them with a named constant
     16. f
     17. f
     18. f
     19. f
     20. f
2. Check order of computation evaluation, operator precedence and parenthesizing
3. Parenthesis to avoid operator precedence
4. No denominators equal to zero

No division

1. Integer arithmetic used appropriately to avoid causing unexpected truncation rounding

No integer arithmetic expression

1. Comparison and Boolean operators correct
2. Throw catch expressions and error condition is actually legitimate

copyResource🡪 1701 line legitimate because if there is no resource in the destination path it hasn’t to delete nothing

1. Code free of any type conversions

No conversion in the code. Only decodification or getter

1. Relevant exceptions are caught
2. Appropriate action taken for each catch block

1701. appropriate action because if it doesn’t exist take into account that no need of deletion.

No switch

1. Switch cases with break or return
2. Switch with default branch
3. All loops well formed : appropriate initialization increment termination expressions

All done from here no file present in lines of code

1. All files declared and opened
2. Closed properly
3. EOF detected and handled correctly
4. File exceptions caught