**Scenarios**

1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Original Code | Explanation | Mutation | Solution |
| 1 | **function** getOpposite(bool) {  **return** !bool; } | Function returns the opposite of a boolean value | Line 2 : return bool; | let bool = false;  const result = getOpposite(bool);  expect(result).toBe(true); |
| 2 | **function** getArea(side1, side2, side3){  **var** perimeter = (side1 + side2 + side3)/2;  **var** area = ***Math***.sqrt(perimeter\*((perimeter-side1)\*(perimeter-side2)\*(perimeter-side3)));  **return** area; } | Function returns the area of a triangle | Line 2: var perimeter = (side1 + side2 - side3)/2; |  |
| 3 | **function** checkSunday (year) {  **for** (**var** y = year; y <= 2050; y++)  {  **var** d = **new *Date***(y, 0, 1);  **if** ( d.getDay() === 0 ){  **return** (**"1st January is being a Sunday "**+year);  }  } } | Function logs every 1st January that is a Sunday between the year set to the year 2050. | Line 2: for (var y = 2014; y <= 2050; y++) |  |
| 4 | **function** guessNumber(gnum){  **var** num = ***Math***.ceil(***Math***.random() \* 10);  **if** (gnum == num) {  **return** (**"Matched"**);}  **else** {  **return** (**"Not matched, the number was "** + num);  }  } | Function lets user guess number between one and ten and matches if correct. | Line 5: Removed  Line 7: Removed |  |
| 5 | **function** setVariable(varName){ **var** n = 120; **this**[varName] = n; **return** (**this**[varName]) } | User chooses variable name and stores the number 120 inside the variable. | Line 2: var n = ‘120’ |  |
| 6 | **function** test50(x, y) {  **if** ((x == 50 || y == 50) || (x + y == 50)){  **return** (**"Success"**);  } } | Checks two inputs If they contain 50 or if the sum of them are 50. | Line 3 : if ((x != 50 || y == 50) || (x + y == 50)){ |  |
| 7 | **function** start\_spec\_str(str) {  **if** (str.**length** < 4)  {  **return false**;  }  **front** = str.substring(0, 4);  **if** (**front** == **"Java"**)  {  **return true**;  }  **else** {  **return false**;  } } | Function checks if a string starts with 'Java' and false otherwise. | Line 12 – 15 : Removed |  |
| 8 | **function** capital\_letter(str) {  str = str.split(**" "**);  **for** (**var** i = 0, x = str.length; i < x; i++) {  str[i] = str[i][0].toUpperCase() + str[i].substr(1);  }  **return** str.join(**" "**); } | Function capitalizes the first letter of each word of a given string | Line 4: for (var i = 0, x = str.length; i > x; i++) { |  |
| 9 | **function** first\_half (str) {  **if** (str.**length** % 2 == 0) {  **return** str.slice(0, str.**length** / 2);  }  **return** str; } | Function extracts the first half of a string of even length | Line 3: return str.slice(0, str.length); |  |
| 10 | **function** sumn(val) {  **var** sn = 0;  **var** i = 0;  **while** (sn <= val) {  sn += i++;  }  **return** i - 2; } | Function that finds the maximum integer n such that 1 + 2 + ... + n <= a given integer. | Line 5: sn = i++; |  |