*Logic Specification Template*

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | Erick Francisco González Martínez | **Program #** | 4 |

|  |  |
| --- | --- |
| **Class Name** | OutputHandler |

|  |  |
| --- | --- |
| **Method Name** | Double round |

|  |  |
| --- | --- |
| **Parameters** | Double number |

|  |  |  |
| --- | --- | --- |
| If number < 0 | | |
| Return ceil(number – 0.5) | | |
| Return floor(number + 0.5) | | |
| **Method Name** | | String convert |

|  |  |
| --- | --- |
| **Parameters** | Double number |

|  |
| --- |
| string s = to\_string ( round (x \* 100000.0) / 100000.0) |
| int length |
| while (s[s.length() - 1] == '0' && length < s.length()){ |
| length = s.find(".") + 6 |
| } |
|  |
| s.pop\_back(); |
| Return s |

|  |  |
| --- | --- |
| **Method Name** | Void OutputHandler::display |

|  |  |
| --- | --- |
| **Parameters** |  |

|  |
| --- |
| cout << " x = " << results[0] << endl; |
| cout << "dof = " << results[1] << endl; |
| cout << " p = " << convert(results[2]) << endl; |

|  |  |
| --- | --- |
| **Class Name** | InputReader |

|  |  |
| --- | --- |
| **Method Name** | InputReader::InputReader() |

|  |  |
| --- | --- |
| **Parameters** |  |

|  |
| --- |
| Double Error = 0 |
| Results(vector<double>(3, 0)) |
| fileName("") |

|  |  |
| --- | --- |
| **Method Name** | InputReader::InputReader |

|  |  |
| --- | --- |
| **Parameters** | const InputReader &ir |

|  |  |  |
| --- | --- | --- |
| results = ir.results; | | |
| fileName = ir.fileName; | | |
| **Method Name** | | InputReader::handleInput |

|  |  |
| --- | --- |
| **Parameters** |  |

|  |
| --- |
| Cin>>x>>dof; |

|  |  |
| --- | --- |
| **Method Name** | getDistributionGamma |

|  |  |
| --- | --- |
| **Parameters** | double num |

|  |  |  |
| --- | --- | --- |
|  | | |
| if (num == 1) | | |
| return 1 | | |
|  | | |
| if (num == 0.5) | | |
| return 1.77245385091 | | |
|  | | |
| return (num - 1) \* getDistributionGamma(num - 1); | | |
|  | | |
| **Method Name** | getDistributionT |

|  |  |
| --- | --- |
| **Parameters** | double dof, double x |

|  |
| --- |
| part1 = (1 + x^ 2) / dof))^ (-(dof + 1) / 2)); |
| part2 = getDistributionGamma((dof + 1) / 2) / (dof \* pi^(1/2) \* getDistributionGamma (dof / 2)) |
| Return part1\*part2; |

|  |  |
| --- | --- |
| **Method Name** | calculateIteration() |

|  |  |
| --- | --- |
| **Parameters** | double x, double dof, double num\_seg |

|  |
| --- |
| width = x / num\_seg |
| p = 0 |
| it\_num\_seg = 0 |
| it\_width = 0 |
| it\_constant = width / 3 |
| width = x / num\_seg |
|  |
|  |
| while (it\_num\_seg <= num\_seg) |
| { |
| If (it\_num\_seg == num\_seg) |
| Then p += it\_constant \* distrubtionT(it\_width, dof) |
| Else if (it\_num\_seg == 0) |
| Then p += it\_constant \* distrubtionT(0, dof) |
| Else if (it\_num\_seg % 2 == 0) |
| p += 2 \* it\_constant \* distrubtionT(it\_width, dof) |
| Else p += 4 \* it\_constant \* distrubtionT(it\_width, dof) |
| End IF |
| it\_width += width |
| ++it\_num\_seg |
| } |
|  |
| return p |
| } |

|  |  |
| --- | --- |
| **Method Name** | InputReader::storeValues() |

|  |  |
| --- | --- |
| **Parameters** |  |

|  |
| --- |
| p1 = calculateIteration(x, dof, num\_seg) |
| num\_seg \*= 2; |
| p2 = calculateIteration(x, dof, num\_seg) |
| while (abs(p1 - p2) >= maxError) |
| { |
| p2 = p1; |
| num\_seg \*= 2 |
| p1 = calculateIteration(x, dof, num\_seg) |
| } |
| Results[2] = p2; |
| Results[1] = dof; |
| Results[0] = x; |