**Test Cases**

# **Introduction**

## **Purpose**

This document documents test procedures and results for Unified Code Count Tool (UCC) Matlab language support. Requirements, installation procedures and usage instructions are documented in UCC User Manual and Release Notes.

## **References**

## **Definitions, Acronyms and Abbreviations**

# **Test Cases**

## **Test Case <OOPS\_1>**

### ***Test Objectives***

This test case tests the keywords in matlab that come under the Classes & Objects and Enumeration (OOPS Keywords)

### ***Test Description***

*Execute UCC with the input Matlab code file as followed*

-dir <source files directory> UCC\_Matlab\_OOPS\_Keywords\_Test1.m

*Verify UCC output file according to the Pass/Fail criteria.*

### ***Pre-conditions***

*a) The UCC executables is accessible.*

*b) Test directories must be writable*

*c) Matlab source file is accessible.*

### ***Post-conditions***

1. *UCC complete execution without error.*
2. *Output files, MATLAB\_outfile.csv and outfile\_cplx.csv, are created.*

### ***Dependencies***

*See the Pre-condition for any dependencies.*

### ***Assumptions and Constraints***

*None.*

### ***Input Specifications***

*Matlab code file: UCC\_Matlab\_OOPS\_Keywords\_Test1.m*

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %TEST CASE ID: OOPS\_1  %TEST CASE DESCRIPTION :  %This test case has the Classes & Objects and Enumeration keywords from the matlab - OOPS Keywords  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %class keyword  import java.lang.\*;  obj = String('mystring');  class(obj)  %classdef, properties and methods keywords  classdef TensileData  enumeration  No (0)  Yes (1)  Off (0)  On (1)  end  properties  Material = 'carbon steel';  SampleNumber = 0;  Stress  Strain  Modulus  end % properties  methods  function obj = set.Material(obj,material)  if ~(strcmpi(material,'aluminum') ||...  strcmpi(material,'stainless steel') ||...  strcmpi(material,'carbon steel'))  error('Material must be aluminum, stainless steel, or carbon steel')  end  obj.Material = material;  end % set.Material  end% methods  end% classde  exist('matlab/general/mkdir.m')  inferiorto('class\_a')  %ismethod keyword is not included in the keyword list - please note  if ismethod(obj1,'eq') && ismethod(obj2,'eq')  tf = obj1 == obj2;  end  %isobject keyword  isobject(h)  %isprop keyword is not included in the keyword list - please note  if isprop(h,'XDataSource')  set(h,'XDataSource','x')  b = loadobj(a)  methodsview java.awt.MenuItem  sobj = saveobj(obj)  A = subsasgn(A, S, B)  ind = subsindex(A)  B = subsref(A,S)  S = substruct('()',{3,5},'.','field')  superclasses('hgsetget')  superiorto('class\_a') |

### ***Expected Output Specifications***

*File MATLAB\_outfile.csv should contain the following information*

*a) Total Physical SLOC is xxx*

*b) Total Logical SLOC is xxx*

*File outfile\_cplx.csv should contain the following information*

*a) Only 1 Matlab source file is processed and file name is the correction input file according to the input specification.*

*b) The following table gives the expected count of keywords.*

|  |  |
| --- | --- |
| **KEYWORD** | **COUNT** |
| class | 2 |
| ismethod | 2 |
| Others | 1 |

### ***Pass/Fail Criteria***

*If the results match those of the Expected Output Specification, the test has passed.* Otherwise the test has failed.

### ***Test Results***

*[Insert test result – test will be rerun before final delivery]*

# **Test Cases**

## **Test Case <OOPS\_2>**

### ***Test Objectives***

This test case tests the keywords in matlab that come under the handle classes (OOPS Keywords)

### ***Test Description***

*Execute UCC with the input Matlab code file as followed*

-dir <source files directory> UCC\_Matlab\_OOPS\_Keywords\_Test2.m

*Verify UCC output file according to the Pass/Fail criteria.*

### ***Pre-conditions***

*a) The UCC executables is accessible.*

*b) Test directories must be writable*

*c) Matlab source file is accessible.*

### ***Post-conditions***

1. *UCC complete execution without error.*
2. *Output files, MATLAB\_outfile.csv and outfile\_cplx.csv, are created.*

### ***Dependencies***

*See the Pre-condition for any dependencies.*

### ***Assumptions and Constraints***

*None.*

### ***Input Specifications***

*Matlab code file: UCC\_Matlab\_OOPS\_Keywords\_Test2.m*

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %TEST CASE ID: OOPS\_2  %TEST CASE DESCRIPTION :  %This test case has the handle classes keywords from the matlab - OOPS Keywords  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  lh = addlistener(Hsource,'EventName',callback)  P = addprop(Hobj,'PropName')  B = copy(A)  delete(h)  classdef myclass < dynamicprops  Hmatch = findobj(Hobj,<conditions>)  findprop(containers.Map,'Count')  CV = get(H,'PropertyName')  getdisp(H)  classdef MyHandleClass < handle  classdef myclass < hgsetget  Hl = isvalid(Hobj)  notify(Hobj,'EventName')  %relational operators  TF = eq(H1,H2)  TF = ne(H1,H2)  TF = lt(H1,H2)  TF = le(H1,H2)  TF = gt(H1,H2)  TF = ge(H1,H2)  S = set(h)  setdisp(H) |

### ***Expected Output Specifications***

*File MATLAB\_outfile.csv should contain the following information*

*a) Total Physical SLOC is xxx*

*b) Total Logical SLOC is xxx*

*File outfile\_cplx.csv should contain the following information*

*a) Only 1 Matlab source file is processed and file name is the correction input file according to the input specification.*

*b) The following table gives the expected count of keywords.*

|  |  |
| --- | --- |
| **KEYWORD** | **COUNT** |
| classdef | 3 |
| Others | 1 |

### ***Pass/Fail Criteria***

*If the results match those of the Expected Output Specification, the test has passed.* Otherwise the test has failed.

### ***Test Results***

*[Insert test result – test will be rerun before final delivery]*

# **Test Cases**

## **Test Case <OOPS\_3>**

### ***Test Objectives***

his test case tests the keywords in matlab that come under the events,listeners, meta-class and heterogeneous arrays (OOPS Keywords)

### ***Test Description***

*Execute UCC with the input Matlab code file as followed*

-dir <source files directory> UCC\_Matlab\_OOPS\_Test3.m

*Verify UCC output file according to the Pass/Fail criteria.*

### ***Pre-conditions***

*a) The UCC executables is accessible.*

*b) Test directories must be writable*

*c) Matlab source file is accessible.*

### ***Post-conditions***

1. *UCC complete execution without error.*
2. *Output files, MATLAB\_outfile.csv and outfile\_cplx.csv, are created.*

### ***Dependencies***

*See the Pre-condition for any dependencies.*

### ***Assumptions and Constraints***

*None.*

### ***Input Specifications***

*Matlab code file: UCC\_Matlab\_OOPS\_Keywords\_Test3.m*

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %TEST CASE ID: OOPS\_3  %TEST CASE DESCRIPTION :  %This test case has the events,listeners, meta-class and heterogenous arrays keywords from the matlab - OOPS Keywords  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  event.EventData  lh = event.listener(Hobj,'EventName',@CallbackFunction)  event.PropertyEvent  lh = event.proplistener(Hobj,Properties,'PropEvent',@CallbackFunction)  events('handle')  %meta-class keywords  obmeta = meta.class.fromName('MyClass');  mcls = meta.class.fromName(clname);  meta.DynamicProperty  meta.EnumeratedValue  meta.event  meta.MetaData  >> class(m(2))  meta.method  meta.package  mev = meta.package.fromName('event');  P = meta.package.getAllPackages  meta.property  mc = metaclass(obj);  %heterogenous arrays  C = cat(dim,A,B)  C = horzcat(A1,A2,...)  classdef HierarchyRoot < matlab.mixin.Heterogeneous  C = vertcat(A1,A2,...) |

### ***Expected Output Specifications***

*File MATLAB\_outfile.csv should contain the following information*

*a) Total Physical SLOC is xxx*

*b) Total Logical SLOC is xxx*

*File outfile\_cplx.csv should contain the following information*

*a) Only 1 Matlab source file is processed and file name is the correction input file according to the input specification.*

*b) The following table gives the expected count of keywords.*

|  |  |
| --- | --- |
| **KEYWORD** | **COUNT** |
| Meta.class.fromName | 3 |
| Others | 1 |

### ***Pass/Fail Criteria***

*If the results match those of the Expected Output Specification, the test has passed.* Otherwise the test has failed.

### ***Test Results***

*[Insert test result – test will be rerun before final delivery]*

# **Appendix A – Authors List**

*[The following table shows authors and their corresponding test cases]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Author Name** | **Contact Email** | **Test Case ID** | **Notes** |
| Sriranjani Babu | [sriranjb@usc.edu](mailto:sriranjb@usc.edu) | OOPS\_1 | Test case and Test case document preparation |
| Sriranjani Babu | [sriranjb@usc.edu](mailto:sriranjb@usc.edu) | OOPS\_2 | Test case and Test case document preparation |
| Sriranjani Babu | [sriranjb@usc.edu](mailto:sriranjb@usc.edu) | OOPS\_3 | Test case and Test case document preparation |
| Sriranjani Babu | [sriranjb@usc.edu](mailto:sriranjb@usc.edu) | 3DVisualization\_4 | Test case and Test case document preparation |
|  |  |  |  |
|  |  |  |  |