REFERENCES

[1]. Progress 4GL - fourth generation language. (2014). *What is Progress 4GL?* [Online]. Available: http://www.progress4gl.com/index.html. [Accessed: February 23, 2015].

[2]. Progress Software. (2015). *Who We Are - History* [Online]. Available: https://www.progress.com/company-info/who-we-are/history. [Accessed: February 23, 2015].

[3]. Progress Software. (2013, August 17). *4GL Languages* [Online]. Available: http://www.progress.com/en/4gl-languages.html. [Accessed: February 25, 2015].

[4]. Progress Software. (2013, August 17). *Business Application Integration* [Online]. Available: http://www.progress.com/en/business-application-integration.html. [Accessed: February 25, 2015].

[5]. Wikipedia. (2004, February 6). *Fourth-generation programming language* [Online]. Available: http://en.wikipedia.org/wiki/Fourth-generation_programming_language. [Accessed: March 07, 2015].

[6]. IBM Software. (2014). IBM® Informix® 4GL [Online]. Available: http://www-03.ibm.com/software/products/en/4gl/. [Accessed: February 2, 2014].

[7]. Progress Software. (2014). *Progress Software* [Online] Available: http://progress4gl.com/. [Accessed: June 30, 2014].

[8]. IBM Software, “Transitioning: Informix 4GL To Enterprise Generation Language EGL” in *IBM Press*, 2005.

[9]. C. Nagy, L. Vidács, R. Ferenc, T. Gyimóthy, F. Kocsis, and I. Kovács. "Solutions for reverse engineering 4GL applications, recovering the design of a logistical wholesale system." in *Software Maintenance and Reengineering (CSMR)*, 2011 15th European Conference on, pp. 343-346. IEEE, 2011.

[10]. M. Rahgozar and F. Oroumchian. "A Transformational Approach for Legacy Systems’ evolution." in *Submitted for publication in: 2002 WSEAS International Conference on Applied Mathematics and Computer Science (AMCOS’02)*, Copacabana, Rio De Janeiro. 2002.

[11]. B. Wilson and T. Van der Beken. "Observations on automation in cross-platform migration." in *ELISA workshop*, p. 140.

[12]. J. C. Riquelme, M. Polo, J. S. Aguilar–Ruiz, M. Piattini, F. J. Ferrer–Troyano, and F. Ruiz. "A comparison of effort estimation methods for 4GL

programs: experiences with Statistics and Data Mining." in *International Journal of Software Engineering and Knowledge Engineering* 16, no. 01, pp. 127-140, 2006.

[13]. Ispirer Systems Ltd. (1999-2014). *Migrate Informix to Oracle* [Online]. Available: http://www.ispirer.com/products/informix-to-oracle-migration. [Accessed: June 30, 2014].

[14]. IBM Information Management Software. (2014). *Converting IBM Informix 4GL Applications to Informix Genero* [Online] Available: ftp://public.dhe.ibm.com/software/data/sw-library/informix/whitepapers/I4GL-to-Genero-Conversion.pdf. [Accessed: June 30, 2014].

[15]. IBM Software. (2011, March). *Infomix Genero* [Online]. Available: http://www-03.ibm.com/software/products/en/infogene/. [Accessed: June 30, 2014].

[16]. MoreData. *The 4GL Connector White Paper.* Informix an IBM trademark 2001-2014.

[17]. MoreData. (2001-2014). *INFORMIX specialized services* [Online]. Available: http://www.moredata.eu/offer/informix.html. [Accessed: June 30, 2014].

[18]. FreeSoft Inc. (2012). *Legacy Modernization* [Online]. Available: http://www.freesoftus.com/en/solutions/legacy_modernization/. [Accessed: June 30, 2014].

[19]. R. Nicola, “Conversion to SAS Software from other 4GL Languages”, *Quality Partner Forum*, SPS Software Services Inc. Canton, Ohio.

[20]. Raincode SPRL. "Automatic migration of your Progress 4GL application to Java", *in EU Gateway Program*, Belgium.

[21]. Global Code Development Corporation, “Progress 4GL to Java Conversion – For Independent Software Vendors”, *for Golden Code Development Corporation Documentation*, 2006-2007.

[22]. J. V. Harrison, and W. M. Lim. "Automated reverse engineering of legacy 4GL information system applications using the ITOC workbench." in *Advanced Information Systems Engineering*, pp. 41-57. Springer Berlin Heidelberg, 1998.

[23]. C. Nagy, L. Vidács, R. Ferenc, T. Gyimóthy, F. Kocsis, and I. Kovács. "Solutions for reverse engineering 4GL applications, recovering the design of a logistical wholesale system." in *Software Maintenance and Reengineering (CSMR)*, 2011 15th European Conference on, pp. 343-346. IEEE, 2011.

[24]. M. Rahgozar and F. Oroumchian. "A Transformational Approach for Legacy Systems’ evolution." in *Submitted for publication in: 2002 WSEAS International Conference on Applied Mathematics and Computer Science (AMCOS’02)*, Copacabana, Rio De Janeiro. 2002.

[25]. B. Wilson and T. Van der Beken. "Observations on automation in cross-platform migration." in *ELISA workshop*, pp. 140.

[26]. R. Ferenc, Á. Beszédes, M. Tarkiainen, and T. Gyimóthy, “Columbus–Reverse Engineering Tool and Schema for C++,” in *Proceedings of the 18th International Conference on Software Maintenance (ICSM’02)*.IEEE Computer Society, pp. 172–181, 2002.

[27]. J. C. Riquelme, M. Polo, J. S. Aguilar–Ruiz, M. Piattini, F. J. Ferrer–Troyano, and F. Ruiz. "A comparison of effort estimation methods for 4GL programs: experiences with Statistics and Data Mining." in *International Journal of Software Engineering and Knowledge Engineering* 16, no. 01, pp. 127-140, 2006.

[28]. K. K. Aggarwal, Y. Singh, P. Chandra, and M. Puri. "Bayesian regularization in a neural network model to estimate lines of code using function points." in *Journal of Computer Science* 1, no. 4, pp. 505, 2005.

[29]. L. L. Minku and X. Yao. "A principled evaluation of ensembles of learning machines for software effort estimation." in *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, p. 9. ACM, 2011.

[30]. G. E. Wittigand G. R. Finnic. "Using artificial neural networks and function points to estimate 4GL software development effort." in *Australasian Journal of Information Systems* 1, no. 2, 2007.

[31]. Google (2010, Aug 10). *Google Translate: Find out how our translations are created.* [Online]. Available: http://translate.google.com.au/about/. [Accessed: March 07, 2015].

[32]. P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst (2013, Aug 13). *Moses: Statistical Machine Translation System*. [Online]. Available: http://www.statmt.org/moses/?n=Moses.Overview. [Accessed: March 07, 2015].

[33]. Google Cloud Platform (2014, October 10). *Google Prediction API*. [Online]. Available: https://cloud.google.com/prediction/docs/developer-guide

[34]. Oracle Web Determinations. *Create a new language translation for a rulebase.* [Online]. Available: http://docs.oracle.com/html/E24270_01/Content/Languages/Create_new_language_translation_for_rulebase.htm/. [Accessed: March 07, 2015].

[35]. R. D. Brown, “Adding linguistic knowledge to a lexical example-based translation system” in *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pp. 22-32. 1999.

[36]. Y. Yang, G. C. J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu. “Learning approaches for detecting and tracking news events” in *IEEE Intelligent Systems* 14, no. 4 (1999): 32-43.

[37]. A. B. Phillips and R. D. Brown. “Cunei machine translation platform: System description.” in *3rd Workshop on Example-Based Machine Translation*. 2009.

[38]. R. D. Brown, “Example-based machine translation in the pangloss system.” in *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pp. 169-174. Association for Computational Linguistics, 1996.

[39]. A. B. Phillips and R. D. Brown. “Training machine translation with a second-order taylor approximation of weighted translation instances.” in *MT Summit*. 2011.

[40]. C. Monson, A. F. Llitjós, R. Aranovich, L. Levin, R. D. Brown, E. Peterson, J. Carbonell, and A. Lavie. “Building NLP systems for two resource-scarce indigenous languages: Mapudungun and Quechua.” *Strategies for developing machine translation for minority languages* (2006): 15.

[41]. R. Gangadharaiah, R. D. Brown, and J. Carbonell. “Spectral clustering for example based machine translation.” in *Proceedings of the Human Language Technology Conference of the NAACL*, Companion Volume: Short Papers, pp. 41-44. Association for Computational Linguistics, 2006.

[42]. Code Project – Anshul (2010, August 10). *Develop Your Own Language Translation System*. [Online]. Available: http://www.codeproject.com/Articles/100126/Develop-Your-Own-Language-Translation-System. [Accessed: March 19, 2015]

[43]. S. Eiichiro, H. Iida, and H. Kohyama. "Translating with examples: a new approach to machine translation." in *The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, no. 3, pp. 203-212. 1990.

[44]. P. Ciancarini,M. J. Wooldridge et.al, “Collection of Revised papers in Agent-Oriented Software Engineering” in *First International Workshop, AOSE* 2000, Limerick, Ireland, June 10, 2000

[45]. M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn. “Rule-based translation with statistical phrase-based post-editing” in *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic, June 2007. Association for Computational Linguistics

[46]. J. McDermott. "R1: A rule-based configurer of computer systems." in *Artificial intelligence* 19, no. 1, pp. 39-88, 1982.

[47]. Hayes-Roth, Frederick. “Rule-based systems” in *Communications of the ACM* 28, no. 9, pp. 921-932, 1985.

[48]. P. F.Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. C. Lai, and R. L. Mercer. "Method and system for natural language translation." U.S. Patent 5,477,451, issued December 19, 1995.

[49]. I. Fliege, R. Grammes, and C. Weber. "ConTraST–a configurable SDL transpiler and runtime environment." in *System Analysis and Modeling: Language Profiles*, Springer Berlin Heidelberg, pp. 216-228, 2006.

[50]. P. Moreau, C. Ringeissen, and M. Vittek. "A pattern-matching compiler." In *Electronic Notes in Theoretical Computer Science* 44, no. 2, pp. 161-180, 2001.

[51]. S. I.Feldman, "A Fortran to C converter," in *ACM SIGPLAN FORTRAN Forum*, vol. 9, no. 2, pp. 21-22. ACM, 1990.

[52]. R. L.London,J. V. Guttag, J. J. Horning, B. W. Lampson, J. G. Mitchell, and G. J. Popek, *Proof rules for the programming language Euclid*, Springer Berlin Heidelberg, 1978, pp 1-26.

[53]. L.A. Hayden, "System and method for generating target language code utilizing an object oriented code generator," U.S. Patent 6,877,155, issued April 5, 2005.

[54]. J. V. Harrison, A. Berglas, and I. Peake, "Legacy 4GL application migration via knowledge-based software engineering technology: a case study," in *Software Engineering Conference*, 1997. Proceedings. 1997 Australian, pp. 70-78. IEEE, 1997.

[55]. M. Rahgozar and F. Oroumchian, "An effective strategy for legacy systems evolution," *Journal of Software Maintenance and Evolution: Research and Practice* 15, no. 5, pp. 325-344, 2003.

[56]. Bauer, C. and King, G., *Hibernate In Action, Practical Object/Relational Mapping*, Manning Publications, 2004.

[57]. Bauer, Christian, and Gavin King. *Java Persistance with Hibernate.* Dreamtech Press, 2006.

[58]. S. J. Russell, P. Norvig, J. F. Canny, Jitendra M. Malik, and Douglas D. Edwards. *Artificial intelligence: a modern approach.* Vol. 74. Englewood Cliffs: Prentice hall, 1995.

[59]. J. F. Allen, *Natural Language Processing.* [Online]. Available: http://ai.ato.ms/MITECS/Entry/allen.html. [Accessed: March 16, 2015]

[60]. R. Wolniewicz, "Auto-Coding and Natural Language Processing." in *3M Health Information Systems*, 2011.

[61]. MIT Technology Review (2013, April 23), *10 Breakthrough Technologies 2013 – Deep Learning*. [Online]. Available: http://www.technologyreview.com/featuredstory/513696/deep-learning/. [Accessed: February 27, 2015].

[62]. Terrance Parr (2014), *ANTLR – Another Tool for Language Recognition*. [Online]. Available: http://www.antlr.org/index.html. [Accessed: February 27, 2015].

[63]. Terrance Parr (2015, January 15), *Getting Started with ANTLR v4*. [Online]. Available: https://theantlrguy.atlassian.net/wiki/display/ANTLR4/Getting+Started+with+ANTLR+v4. [Accessed: February 27, 2015].

[64]. MongoDB Inc. (2015), *mongoDB - Agile and Scalable.* [Online]. Available: http://www.mongodb.org/. [Accessed: February 27, 2015].

[65]. SourceForge.NET (2015, February 1), *CLIPS – Tool for Building Expert Systems.* [Online]. Available: http://clipsrules.sourceforge.net. [Accessed: February 27, 2015].

[66]. JGraph Ltd. (2015), *mxGraph is the market leading JavaScript Graph Visualization Component*. [Online]. Available: https://www.jgraph.com/javascript-graph-visualization-library.html. [Accessed: February 27, 2015]. [Accessed: February 27, 2015].

[67]. JGraphX Version 3.1.2.2 (2015, February 09), *JGraphX (JGraph 6) User Manual*. [Online]. Available: http://jgraph.github.io/mxgraph/docs/manual_javavis.html. [Accessed: February 27, 2015].

[68]. Dr. Shazzad Hosain (2013, March 24), *CLIPS Tutorial 1*. [PDF]. Available: http://w3.northsouth.edu/php/faculty/shazzad/courses/cse513_cse348_ete333/course_reading/CLIPS%20tutorial.pdf. [Accessed: February 27, 2015].

[69]. Progress Software Corporation (1993 - 2015), *OpenEdge Documentation – Progress OpenEdge Release 11 Master Help System* [HTML]

[70]. S. R. Schach, *Object-oriented and classical software engineering*, 7th ed. New York: WCB/McGraw-Hill, 2007.

[71]. J. Brownlee (2014, July 18). *Java Machine Learning*. [Online]. Available: http://machinelearningmastery.com/java-machine-learning/

[72]. S. Yegulalp (2014, August 4). *5 ways to add machine learning to Java, JavaScript, and more*. [Online]. Available: http://www.javaworld.com/article/2461485/big-data/5-ways-to-add-machine-learning-to-java-javascript-and-more.html. [Accessed: March 07, 2015].

[73]. A. C. Oliver (2014, May 29). *Enjoy Machine Learning with Mahout on Hadoop*. [Online]. Available: http://www.infoworld.com/article/2608418/application-development/enjoy-machine-learning-with-mahout-on-hadoop.html. [Accessed: March 07, 2015].

[74]. BigML Inc. (2012, June 7). *Machine Learning in Java has never been easier!* [Online]. Available: http://blog.bigml.com/2012/06/07/machine-learning-in-java-has-never-been-easier/. [Accessed: March 08, 2015].

[75]. I. Vasilev. *A Deep Learning Tutorial: From Perceptrons to Deep Networks*. [Online]. Available: http://www.toptal.com/machine-learning/an-introduction-to-deep-learning-from-perceptrons-to-deep-networks. [Accessed: March 24, 2015].

[76]. Joanju Proparse (2011, November 27). *Parse your OpenEdge ABL source code*. [Online]. Available: http://www.joanju.com/proparse/index.php. [Accessed: March 15, 2015].

[77]. Abeel, Thomas, Yves Van de Peer, and Yvan Saeys. "Java-ML: A machine learning library." in *The Journal of Machine Learning Research 10* (2009): 931-934.

[78]. J. Green, The Open Edge Hive (2009, August 24). *The Proparse Library*. [Online]. Available: http://www.oehive.org/node/1514. [Accessed: March 20, 2015].

[79]. J. Green, The Open Edge Hive (2006, September 05). *Proparse Java/SWT tree view* [Online]. Available: http://www.oehive.org/node/150. [Accessed: March 11, 2015].

[80]. University of Waikato, *Weka 3: Data Mining Software in Java* [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: March 11, 2015].

[81]. RapidMiner (2014), *RapidMiner – Analytics for Anyone* [Online]. Available: https://rapidminer.com/. [Accessed: March 11, 2015].

[82]. F. Engelen (2013, March 22), *Taking a Look at Java-based Machine Lerning Classification* [Online]. Available: https://blog.codecentric.de/en/2013/03/java-based-machine-learning-by-classification/. [Accessed: March 11, 2015].

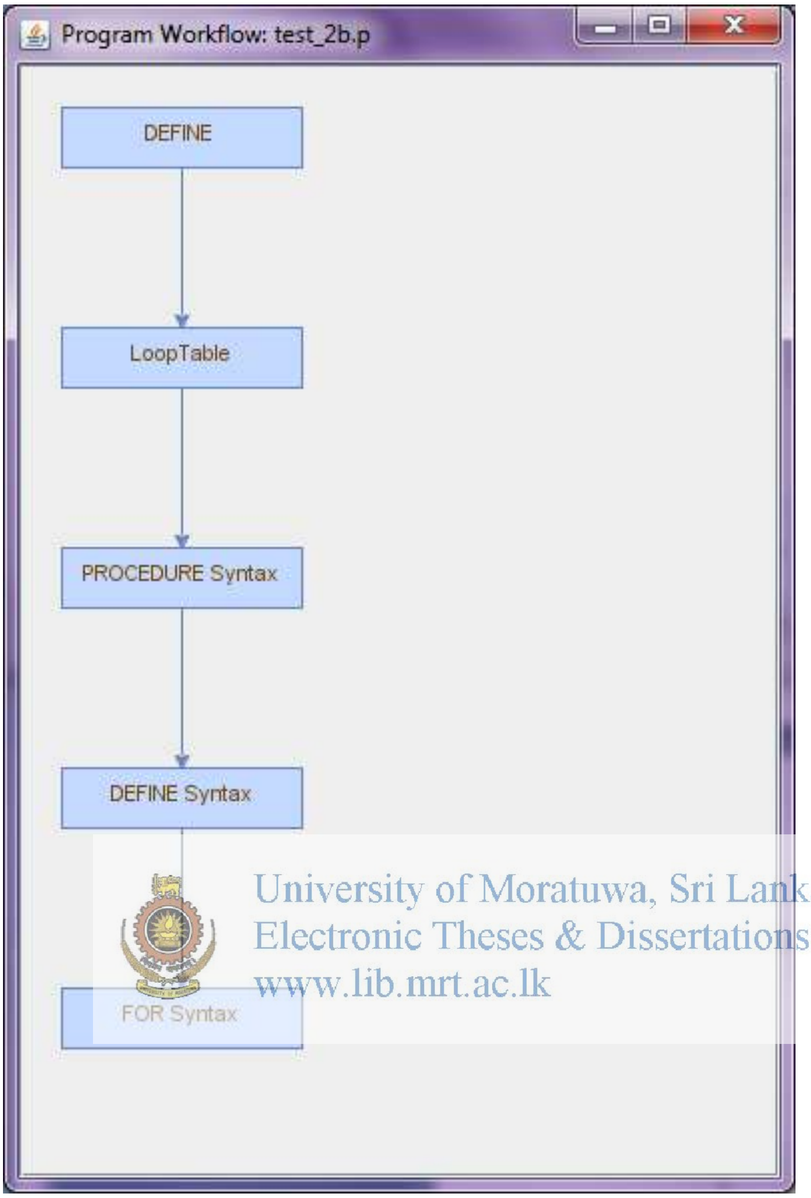
[83]. X. Wang (2013, January), *A Simple Machine Learning Example in Java* [Online]. Available: http://www.programcreek.com/2013/01/a-simple-machine-learning-example-in-java/. [Accessed: March 11, 2015].

[84]. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. "From Data Mining to Knowledge Discovery in Databases." in *AI magazine 17*, no. 3, 1996, pp. 37.

APPENDIX A: 4GL PROGRAM CODE & OUTPUT

- **Sample Program 1**

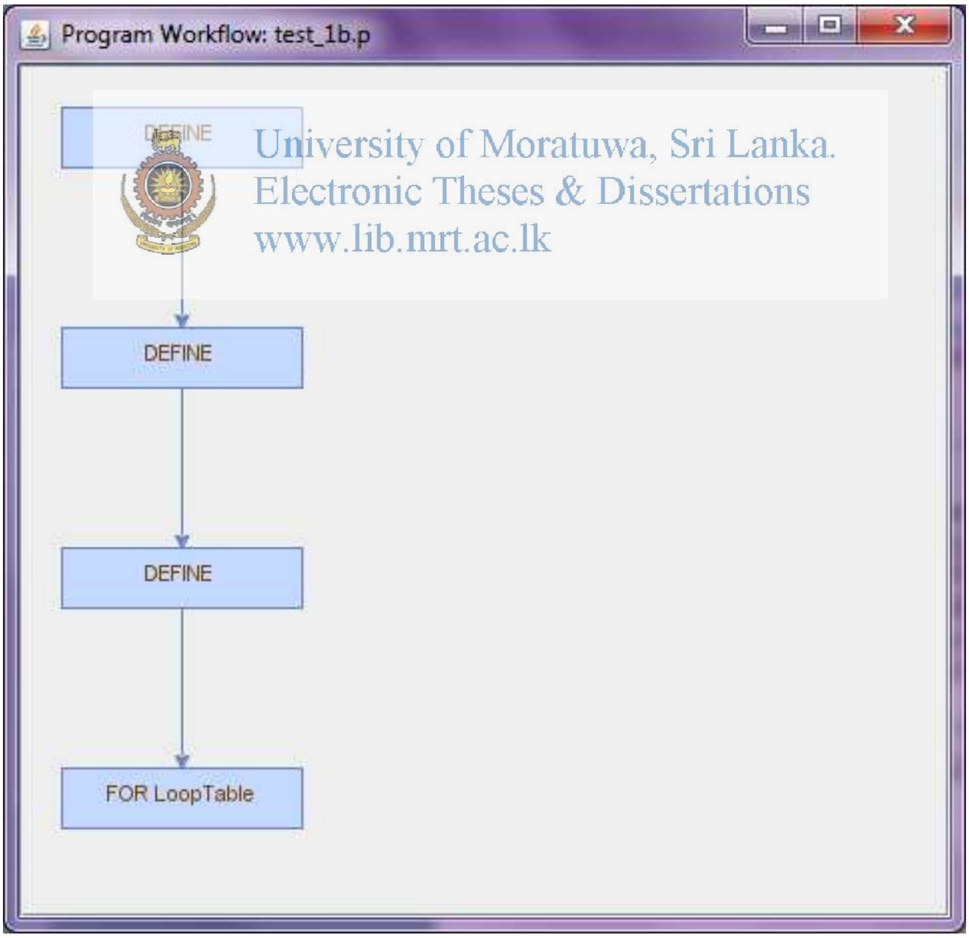
```
/*--------------------------------------------------------------------
   File        : test_2b.p
   Description : Write a call to an internal procedure to calculate the number of orders the customer
placed during this year, and their total value. Display these values as part of (a) above
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 08:40:44 IST 2010
   Notes       :
 ------------------------------------------------------------------*/

/* ****************************  Definitions  *************************** */
DEFINE VARIABLE oCount AS INTEGER LABEL "No of Orders".
DEFINE VARIABLE oTotal AS INTEGER LABEL "Total".

/* ****************************  Main Block  **************************** */
FOR EACH Customer BREAK BY Customer.CustNum:
  RUN NoOfCustOrders(INPUT customer.CustNum, OUTPUT oCount, OUTPUT oTotal).
  DISPLAY Customer.CustNum
          Customer.Name
          Customer.CreditLimit
          oCount
          oTotal
             WITH SIZE 100 BY 100 .

END.


PROCEDURE NoOfCustOrders:
   DEFINE INPUT  PARAMETER ipCustNum LIKE customer.CustNum NO-UNDO.
   DEFINE OUTPUT PARAMETER oCount AS INTEGER.
   DEFIN  OUTPUT PARAMETER oTotal AS INTEGER.

   FOR EACH ORDER
      WHERE ORDER.CustNum        = ipCustNum
       AND YEAR(Order.OrderDate) = 1997:

    FOR EACH ORDERLINE OF ORDER:
      oCount = oCount + 1.
      oTotal = oTotal + (OrderLine.Qty * OrderLine.Price).
      /*ACCUMULATE OrderLine.Qty * OrderLine.Price (TOTAL).
      DISPLAY (ACCUM TOTAL OrderLine.Qty * OrderLine.Price) LABEL "Total".*/
    END.
   END.
END.
```



- **Sample Program 2**

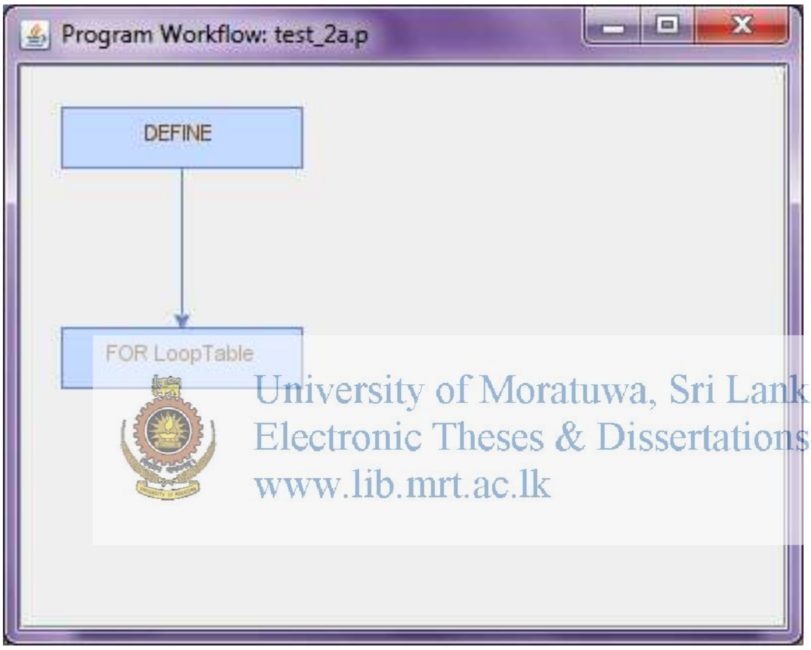
```
/*----------------------------------------------------------------------
   File        : test_1b.p
   Description : Modify the above code to delete all records where the credit limit is less than 100 and
there are no orders placed. Both tasks should be performed within the same loop.
   Author(s)   : ISubasinghe
   Created     : Tue Mar 16 18:23:23 IST 2010
   Notes       :
 ----------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE VARIABLE cNum   LIKE Customer.CustNum NO-UNDO.
DEFINE VARIABLE cState LIKE Customer.State    NO-UNDO.

DEFINE FRAME frmHead
```

```
   HEADER
    "Customers in " cState
    WITH NO-LABEL.


/* ***************************  Main Block  *************************** */

FOR EACH Customer BREAK BY Customer.State:
   IF FIRST-OF(Customer.State) THEN
         DISPLAY Customer.State WITH FRAME frmHead NO-LABEL.
   DISPLAY Customer.CustNum
         Customer.Name
         Customer.CreditLimit.

   FIND FIRST Order OF CUSTOMER NO-LOCK NO-ERROR.
   IF AVAILABLE(Order) AND Customer.CreditLimit < 20000 THEN
   DO:
      ASSIGN cNum = Customer.CustNum.
      DELETE Customer.
      MESSAGE "CUSTOMER " cNum " DELETED".

   END.
END.
```



- **Sample Program 3**

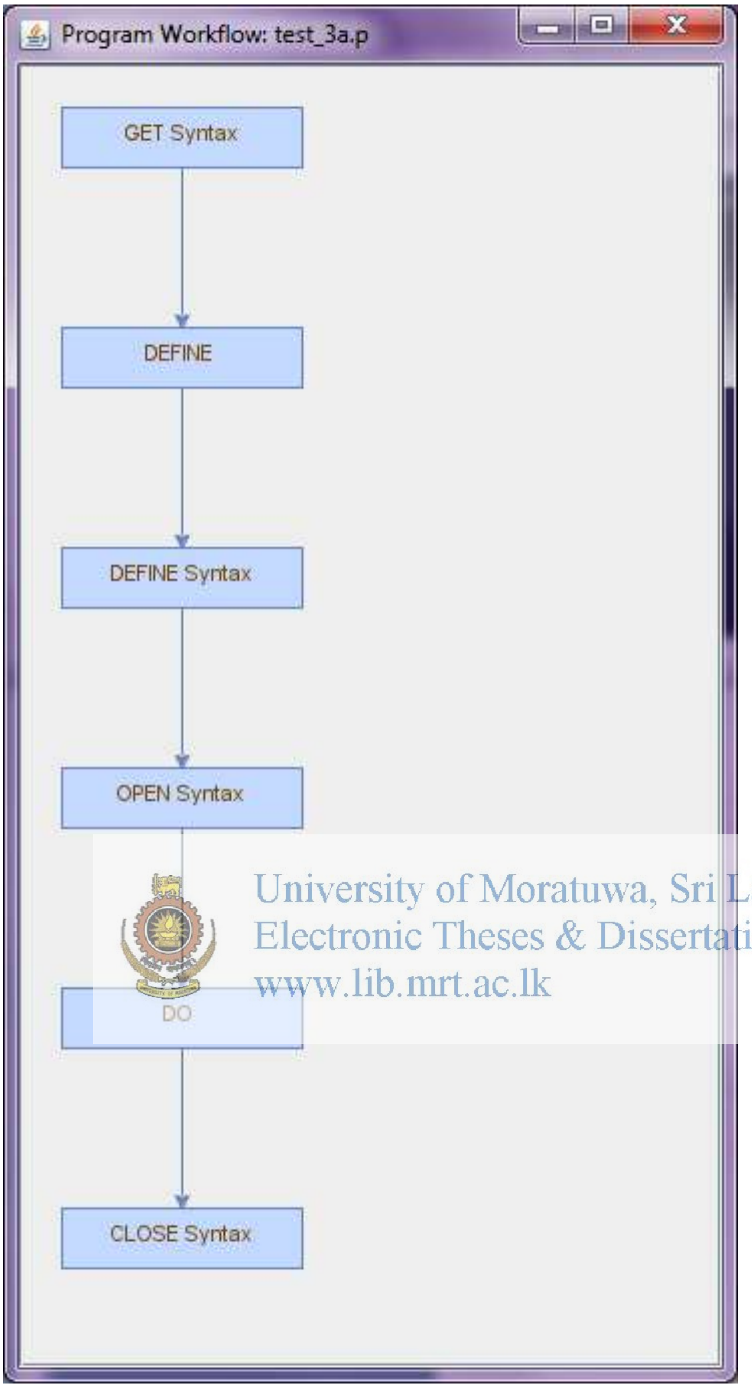
```
/*------------------------------------------------------------------------
   File        : test_2a.p
   Description : Write a procedure which goes through the customer table and displays customer code, name and credit limit.
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 08:18:32 IST 2010
   Notes       :
 ----------------------------------------------------------------------*/

/* ***************************  Main Block  *************************** */
FOR EACH Customer:
   DISPLAY Customer.CustNum
         Customer.Name
         Customer.CreditLimit.
END.
```



- **Sample Program 5**

```
/*------------------------------------------------------------------------
   File        : test_3a
   Description : Use a query to display the customer code, customer name, and the number of customer results at each line.
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 10:36:19 IST 2010
   Notes       :
 ----------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE QUERY    q1  FOR Customer SCROLLING.
DEFINE VARIABLE cnt AS INT.

/* ***************************  Main Block  *************************** */
```

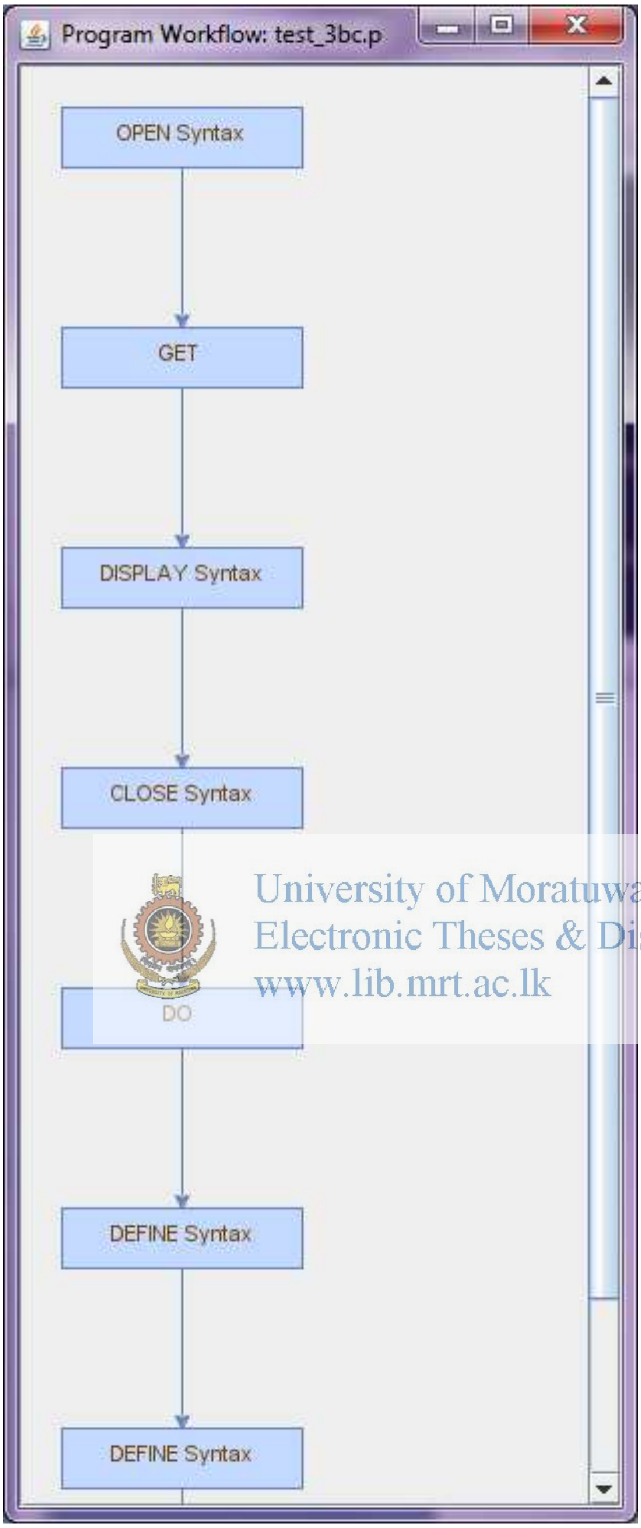
```
OPEN QUERY q1 FOR EACH Customer.

GET FIRST q1 NO-LOCK.

cnt = NUM-RESULTS ("q1").

DO WHILE NOT QUERY-OFF-END ("q1")WITH FRAME DEFAULT-FRAME:

   DISPLAY Customer.CustNum
         Customer.Name
         cnt LABEL "No Of Results".
   GET NEXT q1 NO-LOCK.
   cnt = NUM-RESULTS ("q1").
END.
CLOSE QUERY q1.
```



Program Workflow: test_3a.p
GET Syntax
DEFINE
DEFINE Syntax
OPEN Syntax
DO
CLOSE Syntax

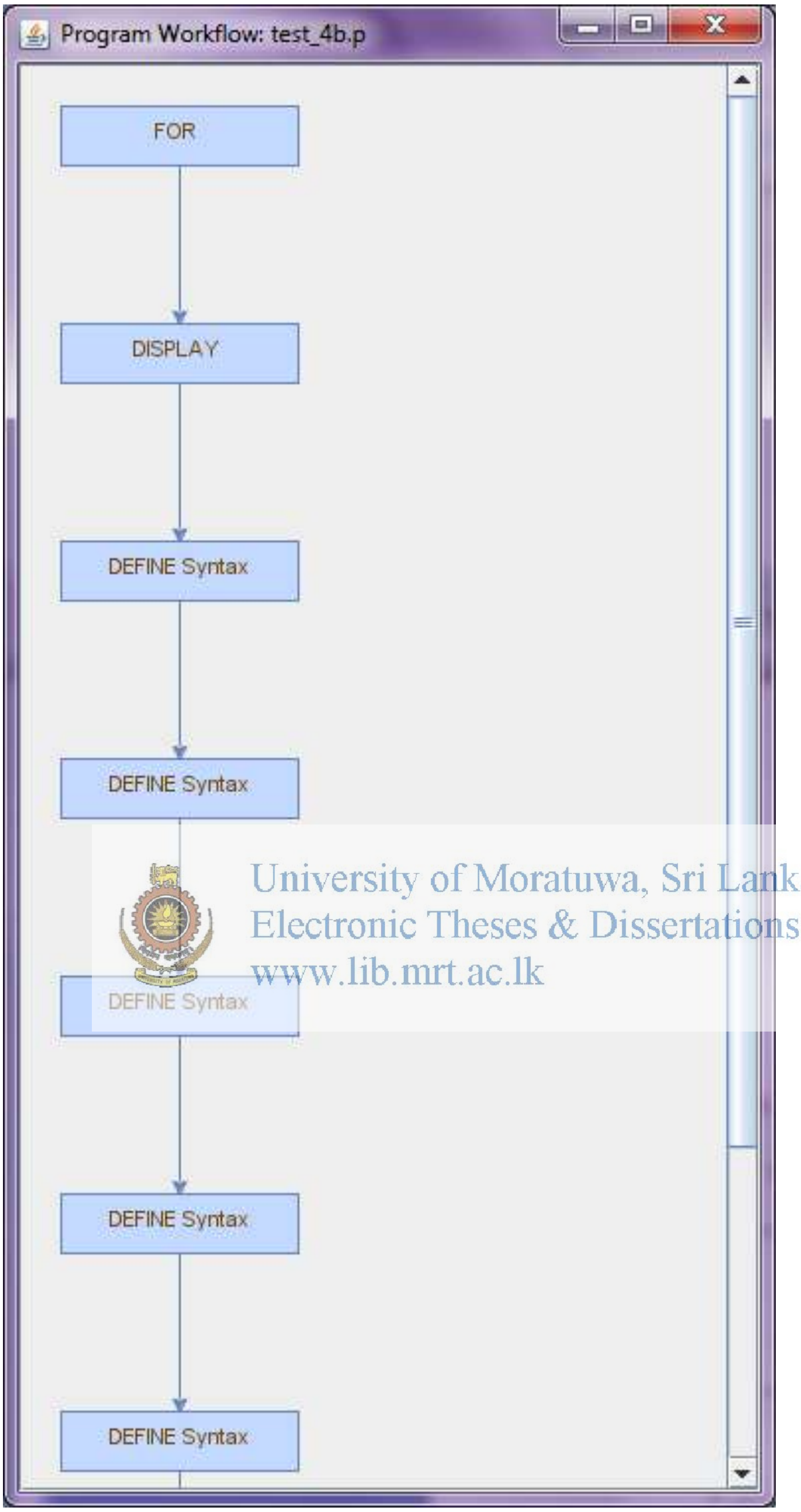
- **Sample Program 6**

```
/*------------------------------------------------------------------------
   File        : test_3b.p
   Description : b. Define and open a query that will preselect all customers, and all the orders for each
customer, if available.
                 c. Display the customer code and name, the number of orders available per customer, and
the value of these orders.
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 11:45:08 IST 2010
   Notes       :
 ----------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE QUERY q1 FOR Customer, Order SCROLLING.

DEFINE VARIABLE cNum   LIKE Customer.CustNum.
DEFINE VARIABLE oCount AS   INT.

/* ***************************  Main Block  *************************** */
OPEN QUERY q1
    PRESELECT EACH Customer, EACH Order OF Customer.

DISPLAY NUM-RESULTS ("q1") LABEL "No Of Results" WITH FRAME frmHdr.

GET FIRST q1 NO-LOCK.

DO WHILE NOT QUERY-OFF-END ("q1") WITH FRAME DEFAULT-FRAME:

  ASSIGN cNum = Customer.CustNum.
  MESSAGE cNum VIEW-AS ALERT-BOX.
   DO WHILE Customer.CustNum = cNum WITH FRAME DEFAULT-FRAME:
      oCount = oCount + 1.
      GET NEXT q1.
   END.

   GET PREV q1.
   DISPLAY Customer.CustNum
          Customer.Name
          oCount LABEL "No Of Orders".
   oCount = 0.
   GET NEXT q1 NO-LOCK.

END.

CLOSE QUERY q1.
```


Program Workflow: test_3bc.p
OPEN Syntax
GET
DISPLAY Syntax
CLOSE Syntax
DO
DEFINE Syntax
DEFINE Syntax

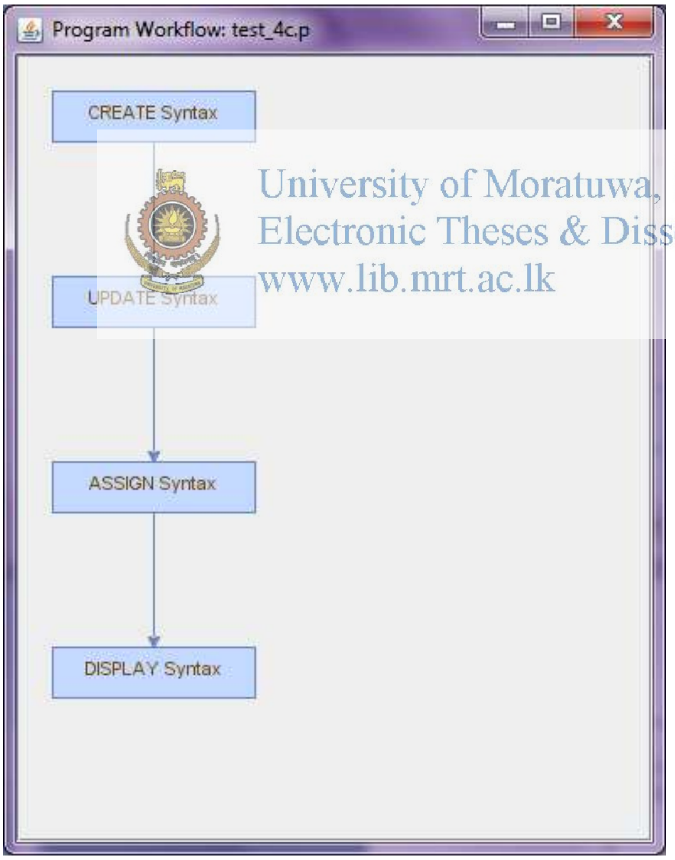
- **Sample Program 7**

```
/*---------------------------------------------------------------------------
   File        : test_4b
   Description : Write a procedure which displays the code and name of all customers who have
invalid orders, along with the order code and error status id
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 14:42:48 IST 2010
   Notes       :
  ------------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE VARIABLE oState     AS CHARACTER.
DEFINE VARIABLE oShipCount AS INT.
DEFINE VARIABLE oOrdCount  AS INT.
DEFINE VARIABLE oBOCount   AS INT.
DEFINE VARIABLE oPSCount   AS INT.
DEFINE VARIABLE oINVCount  AS INT.

/* ***************************  Main Block  *************************** */
FOR EACH Customer, EACH Order OF Customer:
      CASE Order.OrderStatus:
         WHEN "Shipped"          THEN oShipCount = oShipCount + 1.
         WHEN "Ordered"          THEN oOrdCount  = oOrdCount  + 1.
         WHEN "Back Ordered"     THEN oBOCount   = oBOCount   + 1.
         WHEN "Partially Shipped" THEN oPSCount   = oPSCount   + 1.
         OTHERWISE   DO:
                     oINVCount = oINVCount + 1.
                     DISPLAY Customer.CustNum SKIP
                           Customer.Name    SKIP
                           Order.Ordernum   SKIP
                           Order.OrderStatus.
                  END.
      END CASE.
END.

IF oINVCount = 0 THEN
   MESSAGE "No Invalid Orders!" VIEW-AS ALERT-BOX.
```


Program Workflow: test_4b.p
FOR
DISPLAY
DEFINE Syntax
DEFINE Syntax
DEFINE Syntax
DEFINE Syntax
DEFINE Syntax

- **Sample Program 8**

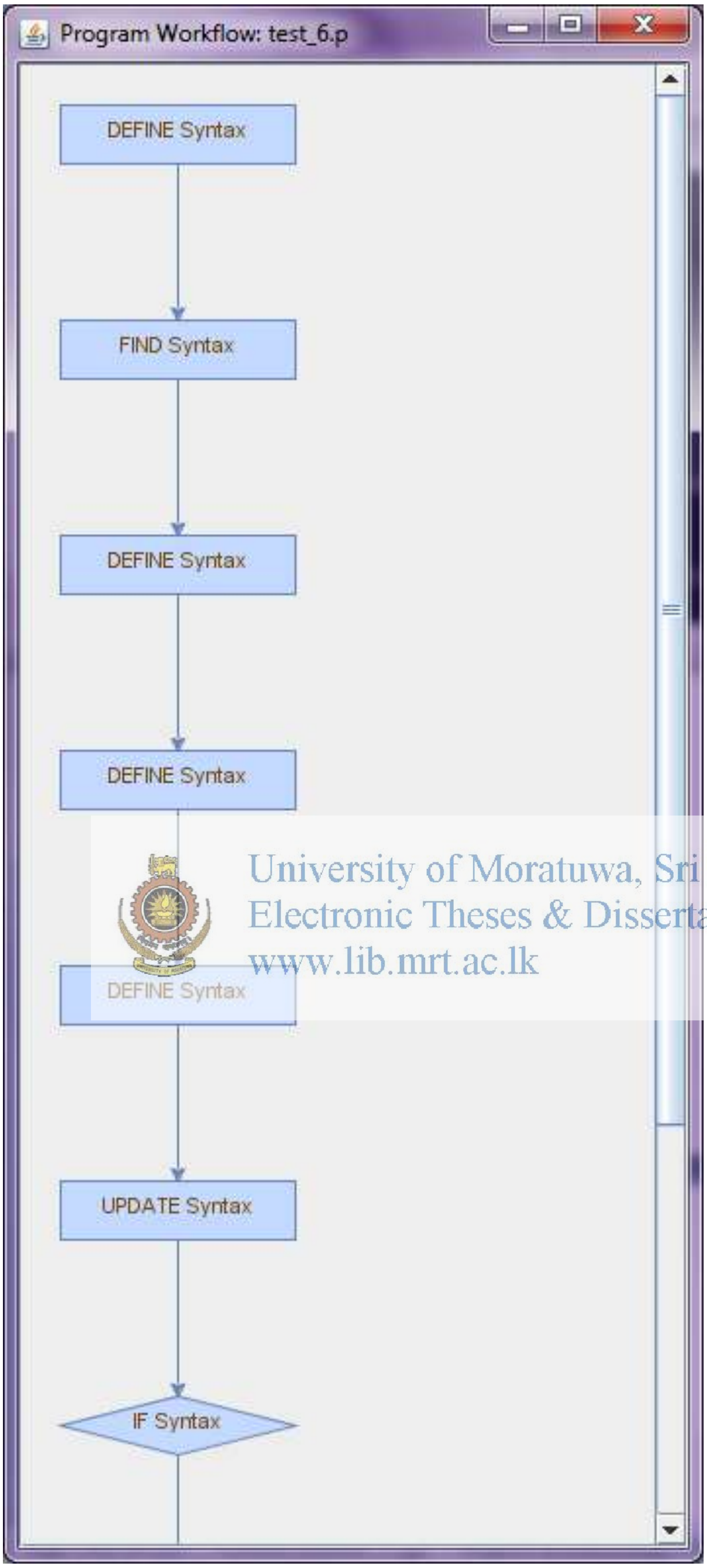
```
/*-----------------------------------------------------------------------
   File        : test_4c.p
   Description : Write a trigger to execute when an order is created, which sets the orderDate to the current date and the deliveryDate to two weeks from the current date
   Author(s)   : ISubasinghe
   Created     : Wed Mar 17 15:16:35 IST 2010
   Notes       :
 ----------------------------------------------------------------------*/

/* ***************************  Main Block  *************************** */
CREATE Order.
UPDATE Order.CustNum.
ASSIGN Order.CustNum.
DISPLAY Order.CustNum
      Order.Ordernum
      Order.OrderDate
      Order.PromiseDate.
```



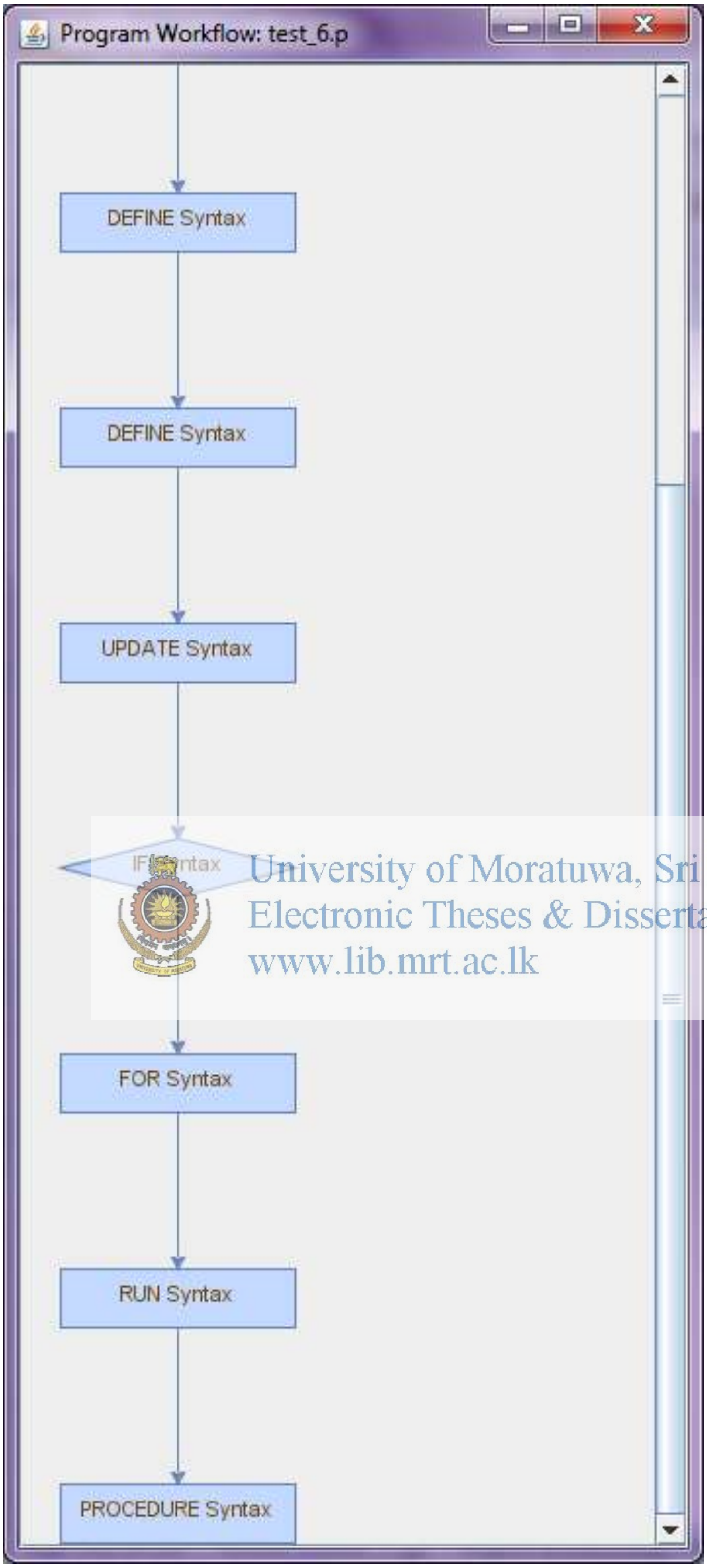
- **Sample Program 9**

```
/*------------------------------------------------------------------------
    File        : test_6.p
    Description :   a. Create temp-tables for the customer, order and orderline tables.
                    b. Write code to accept user input for the customer code, and populate the temp-tables
with related records
                    c. Write the save procedure for the customer temp-table, where the new values are written
back to the
                       database only when the record in the database remains unchanged.
    Author(s)   : ISubasinghe
    Created     : Wed Mar 17 16:55:07 IST 2010
    Notes       :
  ----------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE TEMP-TABLE ttCustomer  LIKE Customer.
DEFINE TEMP-TABLE ttOrder     LIKE Order.
DEFINE TEMP-TABLE ttOrderLine LIKE OrderLine.

DEFINE VARIABLE cCustNum LIKE Customer.CustNum VIEW-AS FILL-IN.

/* ***************************  Main Block  *************************** */
CURRENT-WINDOW:WIDTH-CHARS = 110.
UPDATE cCustNum LABEL "Customer Number".

FIND FIRST Customer WHERE Customer.CustNum = cCustNum NO-ERROR.

IF AVAILABLE(Customer) THEN
DO:
   CREATE ttCustomer.
   BUFFER-COPY Customer TO ttCustomer.

   FOR EACH Order OF Customer WHERE Order.CustNum = cCustNum NO-LOCK:
      CREATE ttOrder.
      BUFFER-COPY Order TO ttOrder.

      FOR EACH OrderLine OF Order NO-LOCK:
         CREATE ttOrderLine.
         BUFFER-COPY OrderLine TO ttOrderLine.

      END.
   END.
   DISPLAY ttCustomer.CustNum ttOrder.Ordernum.
END.
ELSE
DO:
   CREATE ttCustomer.
   UPDATE ttCustomer.Name
        ttCustomer.CreditLimit
        ttCustomer.Country.
   ASSIGN ttCustomer.CustNum = NEXT-VALUE(NextCustNum)
        ttCustomer.Name
        ttCustomer.CreditLimit
        ttCustomer.Country.
END.
```

```
FOR EACH ttCustomer NO-LOCK:
   DISPLAY TTcustomer WITH FRAME f1 WIDTH 100.
END.
RUN saveTtCustomer.


PROCEDURE saveTtCustomer:

   FOR EACH ttCustomer:
      FIND Customer WHERE Customer.CustNum = ttCustomer.CustNum EXCLUSIVE-LOCK NO-ERROR.
         IF CAN-FIND (Customer) THEN

            IF CURRENT-CHANGED (Customer) THEN
               MESSAGE "Customer record has been changed! Cannot Continue to copy."
                     VIEW-AS ALERT-BOX ERROR.
        ELSE
          CREATE Customer.

       BUFFER-COPY ttCustomer TO Customer.
       MESSAGE "Customer " cCustNum " saved!" VIEW-AS ALERT-BOX BUTTON OK.

   END.
END.
```



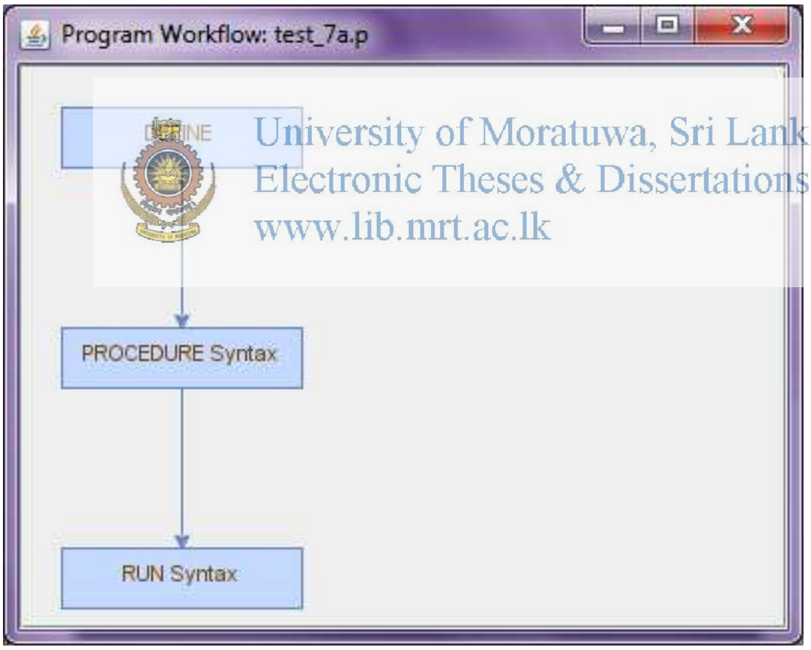
Program Workflow: test_6.p
DEFINE Syntax
FIND Syntax
DEFINE Syntax
DEFINE Syntax
DEFINE Syntax
UPDATE Syntax
IF Syntax



Program Workflow: test_6.p
DEFINE Syntax
DEFINE Syntax
UPDATE Syntax
IF Syntax
FOR Syntax
RUN Syntax
PROCEDURE Syntax

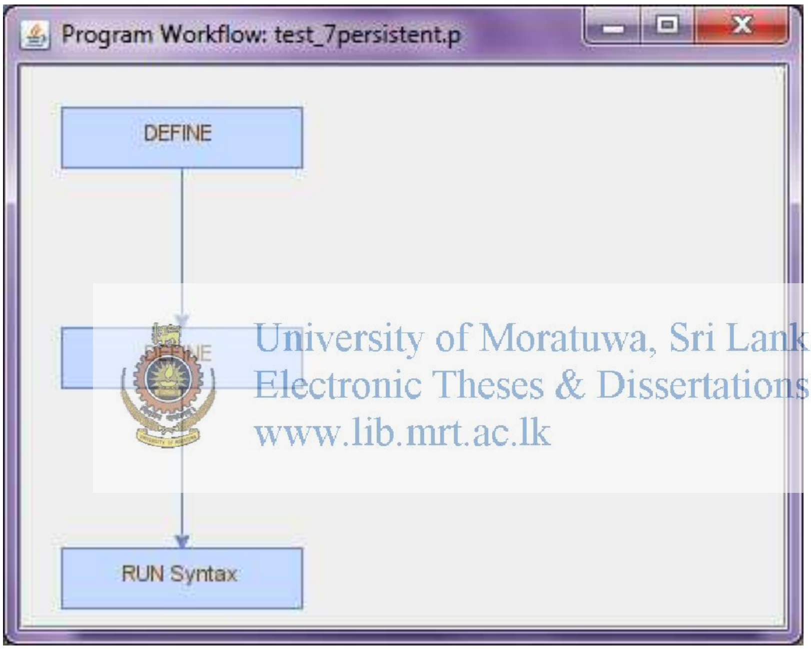
- **Sample Program 10**

```
/*------------------------------------------------------------------------
    File        : test_7a.p
    Description : Write the code for a procedure which calls a persistent procedure, which in turn calls
an internal procedure in the parent to display an error message which is sent in by the persistent
procedure.
    Author(s)   : ISubasinghe
    Created     : Wed Mar 17 18:11:29 IST 2010
    Notes       :
  ----------------------------------------------------------------------*/

DEFINE VARIABLE hParent AS HANDLE.

/* ***************************  Main Block  *************************** */
RUN test_7persistent.p PERSISTENT (INPUT THIS-PROCEDURE).


PROCEDURE displayError:
    DEFINE INPUT PARAMETER msg AS CHARACTER.

    MESSAGE msg VIEW-AS ALERT-BOX ERROR.
END.
```



- **Sample Program 11**

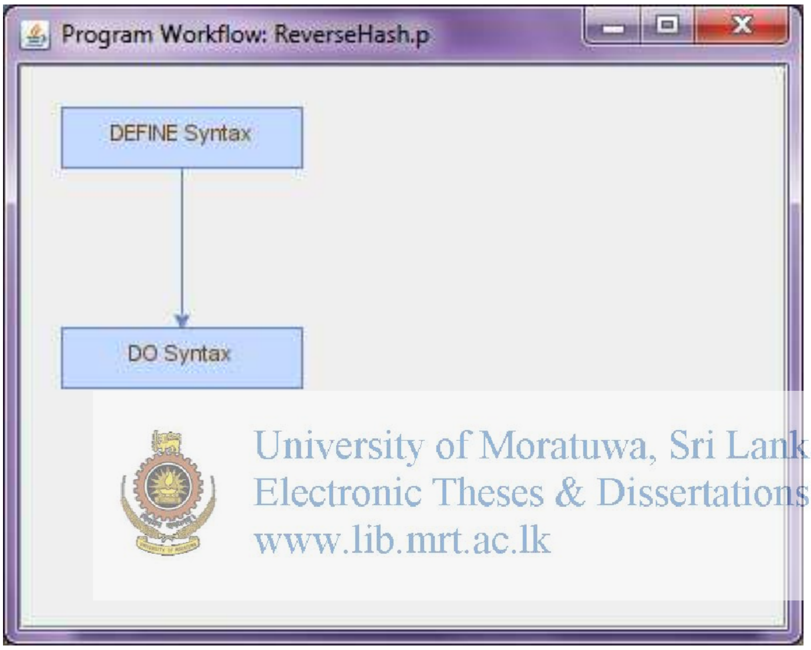
```
/*------------------------------------------------------------------------
    File        : test_7persistent.p
    Author(s)   : ISubasinghe
    Created     : Wed Mar 17 18:22:17 IST 2010
    Notes       :
  ----------------------------------------------------------------------*/

/* ***************************  Definitions  ************************** */
DEFINE VARIABLE errMsg AS CHARACTER INITIAL "My Message from test_7persitent.p!".
DEFINE INPUT PARAMETER hParent AS HANDLE.

RUN displayError IN hParent (INPUT errMsg).
```



- **Sample Program 12**

```
/* ------------------------------------------------------------------- */
/* Program name : ReverseHash.p                                        */
/* Description  : Decrypt hash to return word generated through hash () */
/*              where base value is 7 and multiplier wis 47.          */
/* Syntax       : RUN ReverseHasp.p(INPUT int64HashNum).              */
/* Author       : Ilakshini Subasinghe                                 */
/* Date created : 06-MAR-2015                                          */
/* ------------------------------------------------------------------- */

/* *************** I/O Param *************** */
DEFINE INPUT  PARAMETER ipi64HashValue  AS INT64 NO-UNDO.

/* *************** Forward Declarations *************** */
FUNCTION    fnGetChar    RETURNS    CHAR    (INPUT-OUTPUT    iReverseHash    AS    INT64)
FORWARD.

/* *************** Internal Variables *************** */
DEFINE VARIABLE  i            AS INTEGER                 NO-UNDO.
DEFINE VARIABLE  iReverseHash AS INT64 INITIAL 371580748701652777 NO-UNDO.
DEFINE VARIABLE  word          AS CHARACTER               NO-UNDO.
DEFINE VARIABLE  iBase         AS INTEGER   INITIAL "7"     NO-UNDO.

/* *************** MAIN *************** */

ASSIGN iReverseHash = ipi64HashValue. /* Backup Value */

/* Get each letter from hash (Returns last letter first) */
DO WHILE (iReverseHash > iBase):
   ASSIGN word = fnGetChar(INPUT-OUTPUT iReverseHash) + word.
END.

/* Display Result */
MESSAGE "Your Word Embedded in" ipi64HashValue "is:" SKIP word
   VIEW-AS ALERT-BOX.

/* *************** END MAIN *************** */


/* *************** Function Declarations *************** */
FUNCTION fnGetChar RETURNS CHARACTER (INPUT-OUTPUT iReverseHash AS INT64):
/* ------------------------------------------------------------------- */
/* Function  : fnGetChar                                               */
/* Purpose   : Returns Character Hidden in Hash Number.                */
/* Parameters: i/p: Hash Number                                        */
/* ------------------------------------------------------------------- */
   DEFINE VARIABLE i             AS INTEGER              NO-UNDO.
   DEFINE VARIABLE i64Remainder     AS INT64               NO-UNDO.
   DEFINE VARIABLE cLetters        AS CHARACTER            NO-UNDO.
   DEFINE VARIABLE iBase           AS INTEGER   INITIAL "7"   NO-UNDO.
   DEFINE VARIABLE iMultiplier      AS INTEGER   INITIAL "47"  NO-UNDO.

   ASSIGN
     cLetters = "abcdeghilmnoprstuwy".
```

```
    IF iReverseHash = iBase THEN
       RETURN "".

    DO  i = 1 TO LENGTH(cLetters):
       i64Remainder = iReverseHash - i.
       IF i64Remainder MODULO iMultiplier   = 0 THEN
       DO:
          ASSIGN
             iReverseHash = i64Remainder / iMultiplier.
          RETURN SUBSTRING(cLetters,i,1).
       END.
    END.
END.
```



APPENDIX B: CLASSIFICATION OUTPUT

Rules.NNge (WEKA)

```
=== Run information ===

Scheme:weka.classifiers.rules.NNge -G 5 -I 5
Relation:     4GLkeywords
Instances:    34
Attributes:   4
              token_type_parent
              token_type_child1
              token_type_child2
              mxgraph_vertex_shape
Test mode:evaluate on training data

=== Classifier model (full training set) ===

NNGE classifier

Rules generated :
        class loop IF : token_type_parent in {DO,FOR,PRESELECT,PROMPT-FOR,UPDATE} ^
token_type_child1 in {PRESELECT,TO,EACH,EDITING} ^ token_type_child2 in {FIRST,?}  (6)
        class loop IF : token_type_parent in {DO,FOR,PRESELECT} ^ token_type_child1 in
{FOR,FIRST,LAST,PRESELECT} ^ token_type_child2 in {EACH,LAST}  (8)
        class rhombus IF : token_type_parent in {DISPLAY,DO,PROMPT-FOR,UPDATE,IF} ^
token_type_child1 in {WHEN,QUERY-TUNING,WHILE,CAN-DO} ^ token_type_child2 in {?}  (6)
        class rhombus IF : token_type_parent in {ASSIGN,CASE} ^ token_type_child1 in {WHEN,?}
^ token_type_child2 in {?}  (2)
        class default IF : token_type_parent in
{DISPLAY,DEFINE,DO,FOR,PRESELECT,PROMPT-FOR,UPDATE} ^ token_type_child1 in
{VARIABLE,TEMP-TABLE,BUFFER,FOR,FIRST,LAST,?} ^ token_type_child2 in {FIRST,?}
(12)

Stat :
        class default : 1 exemplar(s) including 1 Hyperrectangle(s) and 0 Single(s).
        class rhombus : 2 exemplar(s) including 2 Hyperrectangle(s) and 0 Single(s).
        class loop : 2 exemplar(s) including 2 Hyperrectangle(s) and 0 Single(s).

        Total : 5 exemplars(s) including 5 Hyperrectangle(s) and 0 Single(s).

        Feature weights : [0.5212479644597324 1.1260791090720865 0.41986782197619227]



Time taken to build model: 0 seconds

=== Predictions on training set ===

inst#,    actual, predicted, error, probability distribution
    1  1:default  2:rhombus      +   0     *1      0
    2  1:default  1:default        *1      0      0
    3  1:default  1:default        *1      0      0
    4  1:default  1:default        *1      0      0
    5  1:default     3:loop      +   0      0     *1
```

```
   6  1:default  1:default       *1      0      0
   7  1:default     3:loop     +  0      0     *1
   8  1:default     3:loop     +  0      0     *1
   9  1:default     3:loop     +  0      0     *1
  10  1:default     3:loop     +  0      0     *1
  11  1:default     3:loop     +  0      0     *1
  12  1:default     3:loop     +  0      0     *1
  13  2:rhombus  2:rhombus          0     *1      0
  14  2:rhombus  2:rhombus          0     *1      0
  15  2:rhombus  2:rhombus          0     *1      0
  16  2:rhombus  2:rhombus          0     *1      0
  17  2:rhombus  2:rhombus          0     *1      0
  18  2:rhombus  2:rhombus          0     *1      0
  19  2:rhombus  2:rhombus          0     *1      0
  20  2:rhombus  2:rhombus          0     *1      0
  21     3:loop     3:loop       0      0     *1
  22     3:loop     3:loop       0      0     *1
  23     3:loop     3:loop       0      0     *1
  24     3:loop     3:loop       0      0     *1
  25     3:loop     3:loop       0      0     *1
  26     3:loop     3:loop       0      0     *1
  27     3:loop     3:loop       0      0     *1
  28     3:loop     3:loop       0      0     *1
  29     3:loop     3:loop       0      0     *1
  30     3:loop     3:loop       0      0     *1
  31     3:loop     3:loop       0      0     *1
  32     3:loop     3:loop       0      0     *1
  33     3:loop     3:loop       0      0     *1
  34     3:loop     3:loop       0      0     *1

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          26               76.4706 %
Incorrectly Classified Instances         8               23.5294 %
Kappa statistic                          0.6334
Mean absolute error                      0.1569
Root mean squared error                  0.3961
Relative absolute error                 36.0976 %
Root relative squared error             85.0462 %
Total Number of Instances               34

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.333     0          1         0.333     0.5        0.667    default
                 1         0.038      0.889     1         0.941      0.981    rhombus
                 1         0.35       0.667     1         0.8        0.825    loop
Weighted Avg.    0.765     0.153      0.837     0.765     0.727      0.806

=== Confusion Matrix ===

  a  b  c   <-- classified as
  4  1  7 |  a = default
  0  8  0 |  b = rhombus
  0  0 14 |  c = loop
```

Bayes.NaiveBayes (WEKA)

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     4GLkeywords
Instances:    34
Attributes:   4
              token_type_parent
              token_type_child1
              token_type_child2
              mxgraph_vertex_shape
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

                    Class
Attribute             default rhombus    loop
                   (0.35)  (0.24)  (0.41)
============================================
token_type_parent
  DISPLAY               2.0     2.0     1.0
  DEFINE               4.0     1.0     1.0
  DO                 3.0     3.0     7.0
  FOR                 3.0     1.0     4.0
  PRESELECT             3.0     1.0     4.0
  PROMPT-FOR              2.0     2.0     2.0
  UPDATE              2.0     2.0     2.0
  ASSIGN              1.0     2.0     1.0
  CASE              1.0     2.0     1.0
  IF                1.0     2.0     1.0
  [total]           22.0    18.0    24.0

token_type_child1
  VARIABLE              2.0     1.0     1.0
  TEMP-TABLE             2.0     1.0     1.0
  BUFFER               2.0     1.0     1.0
  FOR                 2.0     1.0     3.0
  FIRST               3.0     1.0     3.0
  LAST                3.0     1.0     3.0
  WHEN                 1.0     5.0     1.0
  QUERY-TUNING           1.0     2.0     1.0
  WHILE               1.0     2.0     1.0
  CAN-DO               1.0     2.0     1.0
  PRESELECT             1.0     1.0     4.0
  TO                 1.0     1.0     2.0
  EACH                1.0     1.0     3.0
  EDITING              1.0     1.0     3.0
  [total]           22.0    21.0    28.0

token_type_child2
  FIRST               2.0     1.0     2.0
  EACH                1.0     1.0     7.0
  LAST                1.0     1.0     3.0
  [total]            4.0     3.0    12.0
```

```
Time taken to build model: 0 seconds

=== Predictions on training set ===

 inst#     actual  predicted error prediction
     1  1:default  1:default       0.421
     2  1:default  1:default       0.823
     3  1:default  1:default       0.823
     4  1:default  1:default       0.823
     5  1:default     3:loop   +   0.572
     6  1:default  1:default       0.441
     7  1:default     3:loop   +   0.502
     8  1:default     3:loop   +   0.502
     9  1:default     3:loop   +   0.502
    10  1:default     3:loop   +   0.502
    11  1:default     3:loop   +   0.364
    12  1:default     3:loop   +   0.364
    13  2:rhombus  2:rhombus       0.829
    14  2:rhombus  2:rhombus       0.451
    15  2:rhombus  2:rhombus       0.758
    16  2:rhombus     3:loop   +   0.412
    17  2:rhombus     3:loop   +   0.412
    18  2:rhombus  2:rhombus       0.659
    19  2:rhombus  2:rhombus       0.708
    20  2:rhombus  2:rhombus       0.708
    21     3:loop     3:loop       0.81
    22     3:loop     3:loop       0.892
    23     3:loop     3:loop       0.619
    24     3:loop     3:loop       0.646
    25     3:loop     3:loop       0.78
    26     3:loop     3:loop       0.673
    27     3:loop     3:loop       0.72
    28     3:loop     3:loop       0.696
    29     3:loop     3:loop       0.696
    30     3:loop     3:loop       0.72
    31     3:loop     3:loop       0.696
    32     3:loop     3:loop       0.696
    33     3:loop     3:loop       0.569
    34     3:loop     3:loop       0.569

=== Evaluation on training set ===

Time taken to test model on training data: 0.08 seconds

=== Summary ===

Correctly Classified Instances          25               73.5294 %
Incorrectly Classified Instances         9               26.4706 %
Kappa statistic                          0.5785
Mean absolute error                      0.2605
Root mean squared error                  0.314
Relative absolute error                 59.9568 %
Root relative squared error             67.4255 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      91.1765 %
Total Number of Instances               34

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.417    0.000    1.000      0.417    0.588      0.562    0.977     0.972     default
                 0.750    0.000    1.000      0.750    0.857      0.835    1.000     1.000     rhombus
```

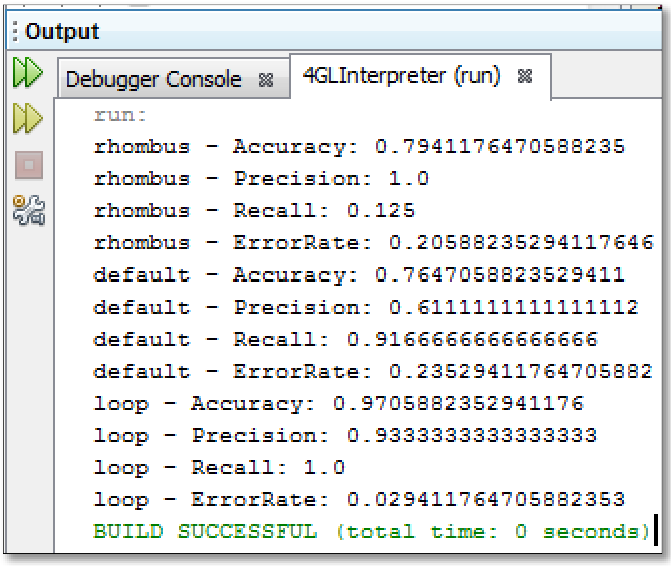
```
              1.000    0.450    0.609     1.000    0.757     0.579    0.993     0.990     loop
Weighted Avg.    0.735    0.185    0.839     0.735    0.721     0.633    0.989     0.986

=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  7 |  a = default
  0  6  2 |  b = rhombus
  0  0 14 |  c = loop
```

Trees.J48 (WEKA)

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     4GLkeywords
Instances:    34
Attributes:   4
              token_type_parent
              token_type_child1
              token_type_child2
              mxgraph_vertex_shape
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
------------------

token_type_child1 = VARIABLE: default (1.17/0.03)
token_type_child1 = TEMP-TABLE: default (1.17/0.03)
token_type_child1 = BUFFER: default (1.17/0.03)
token_type_child1 = FOR: loop (3.52/1.52)
token_type_child1 = FIRST: default (4.69/2.14)
token_type_child1 = LAST: default (4.69/2.14)
token_type_child1 = WHEN: rhombus (4.69/0.55)
token_type_child1 = QUERY-TUNING: rhombus (1.17/0.14)
token_type_child1 = WHILE: rhombus (1.17/0.14)
token_type_child1 = CAN-DO: rhombus (1.17/0.14)
token_type_child1 = PRESELECT: loop (3.52/0.52)
token_type_child1 = TO: loop (1.17/0.17)
token_type_child1 = EACH: loop (2.34/0.34)
token_type_child1 = EDITING: loop (2.34/0.34)

Number of Leaves  :     14

Size of the tree :      15


Time taken to build model: 0.01 seconds

=== Predictions on training set ===

 inst#     actual  predicted error prediction
     1  1:default     3:loop   +   0.412
     2  1:default  1:default       0.971
     3  1:default  1:default       0.971
     4  1:default  1:default       0.971
     5  1:default     3:loop   +   0.412
     6  1:default     3:loop   +   0.569
```

```
    7  1:default  1:default       0.544
    8  1:default  1:default       0.544
    9  1:default  1:default       0.544
   10  1:default  1:default       0.544
   11  1:default     3:loop   +   0.412
   12  1:default     3:loop   +   0.412
   13  2:rhombus  2:rhombus       0.882
   14  2:rhombus     3:loop   +   0.412
   15  2:rhombus  2:rhombus       0.882
   16  2:rhombus  2:rhombus       0.882
   17  2:rhombus  2:rhombus       0.882
   18  2:rhombus  2:rhombus       0.882
   19  2:rhombus  2:rhombus       0.882
   20  2:rhombus  2:rhombus       0.882
   21     3:loop     3:loop       0.569
   22     3:loop     3:loop       0.853
   23     3:loop     3:loop       0.853
   24     3:loop     3:loop       0.569
   25     3:loop     3:loop       0.853
   26     3:loop     3:loop       0.853
   27     3:loop     3:loop       0.853
   28     3:loop  1:default   +   0.544
   29     3:loop  1:default   +   0.544
   30     3:loop     3:loop       0.853
   31     3:loop  1:default   +   0.544
   32     3:loop  1:default   +   0.544
   33     3:loop     3:loop       0.853
   34     3:loop     3:loop       0.853

=== Evaluation on training set ===

Time taken to test model on training data: 0.07 seconds

=== Summary ===

Correctly Classified Instances          24               70.5882 %
Incorrectly Classified Instances        10               29.4118 %
Kappa statistic                          0.543
Mean absolute error                      0.2161
Root mean squared error                  0.3034
Relative absolute error                 49.7226 %
Root relative squared error             65.1435 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      68.6275 %
Total Number of Instances               34

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.583    0.182    0.636      0.583    0.609      0.410    0.852     0.720     default
                 0.875    0.000    1.000      0.875    0.933      0.918    0.990     0.958     rhombus
                 0.714    0.300    0.625      0.714    0.667      0.408    0.954     0.912     loop
Weighted Avg.    0.706    0.188    0.717      0.706    0.709      0.529    0.926     0.855

=== Confusion Matrix ===

  a  b  c   <-- classified as
  7  0  5 |  a = default
  0  7  1 |  b = rhombus
  4  0 10 |  c = loop
```

KNearestNeighbour (Java-ML)



```
run:
rhombus - Accuracy: 0.7941176470588235
rhombus - Precision: 1.0
rhombus - Recall: 0.125
rhombus - ErrorRate: 0.20588235294117646
default - Accuracy: 0.7647058823529411
default - Precision: 0.6111111111111112
default - Recall: 0.9166666666666666
default - ErrorRate: 0.23529411764705882
loop - Accuracy: 0.9705882352941176
loop - Precision: 0.9333333333333333
loop - Recall: 1.0
loop - ErrorRate: 0.029411764705882353
BUILD SUCCESSFUL (total time: 0 seconds)
```