
SUNCOAST
ORACLE
USERS GROUP
WWW.SOUG.NET

Object-Oriented Programming Using PL/SQL

OBJECT-ORIENTED PROGRAMMING (OOP) USING PL/SQL

Mike Kemp

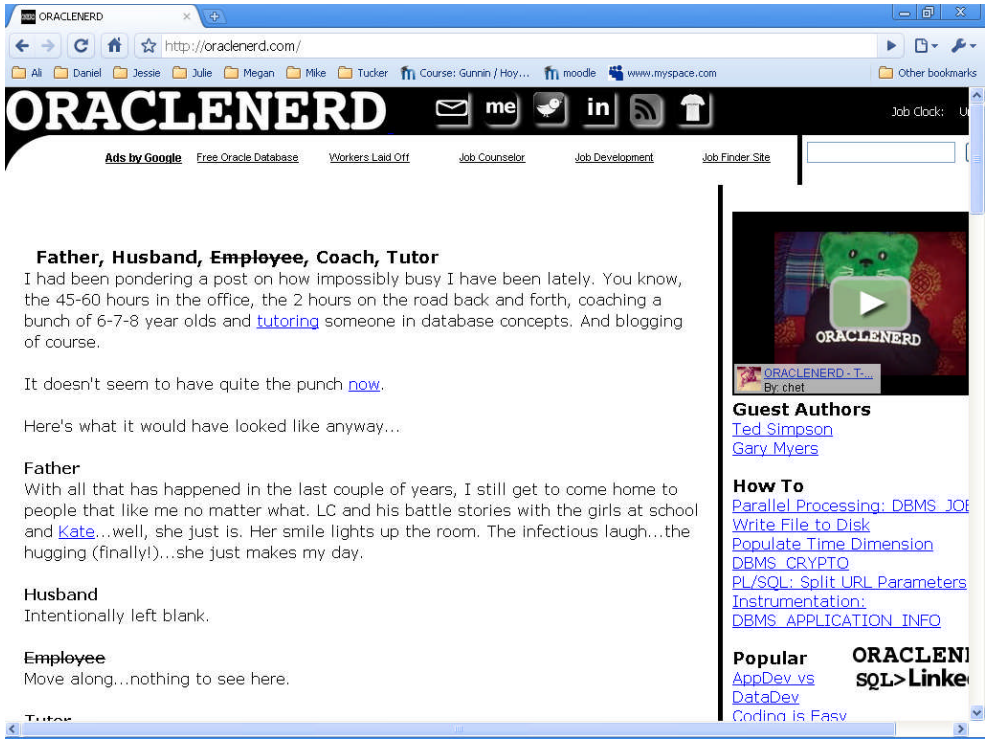
PL/SQL developer
Web developer
Mr. Whatever it takes in IT to stay employed

==========

Tonight
OOP Origins
OOP, Theoretically
OOP and RDBMS?
PL/SQL OOP vs. JAVA OOP
PL/SQL OOP NUTS AND BOLTS
BUT WHY?

eTelligentSolutions – Software License Asset Auto-Discovery, Auto-Redistribution
www.etelligentsolutions.com

Wardrobe By OracleNerd.
Chet Justice, SOUG member
“The Original OracleNerd: Will work for money. Looking for work!!”



Avid blogger.
Great sense of humor.
Estimable Oracle technical skills.
www.oraclenerd.com
CHET JUSTICE

Object-Oriented Programming Using PL/SQL

Group survey:
1) Oracle users? (100%, 99%?)
2) DBA's? (75%?)
3) Developers? (20%?)
4) Object-Oriented Programming basic principles:
    A) Attributes
    B) Methods
    C) Encapsulation
    D) Abstraction
    E) Inheritance
    F) Polymorphism
    G) Type evolution
5) How many DBA's are familiar with these basic principles? (XXX%?)
6) How man Oracle Developers are familiar with these basic principles? (XXX%?)

Object-Oriented Programming Using PL/SQL

About me

== Personal

1) 6 children (4 children, 2 step-daughters). All amazing!!

== Professional

2) IT Professional since 1998.
3) Started working with a database in 1993 when I had my own multi-national cargo carrier: On-Time Courier Service. Heard of it?
4) Can I say this at an Oracle meeting: MS Works?
5) Graduated from MS Works to MS Access to FoxPro to Borland Interbase to SAS and finally and lastly (I promise!) to Oracle.
6) Financial services industry for majority of 11 years. Always mortgages.

== Personal

7) Known for making big, grandiose statements (yeah, it gets me in trouble sometimes).
8) Have been accused of being a dreamer.
9) NOT an expert in OOP.
   A) Interested in the trend among database languages.
   B) Studying up on the subject and experimenting feverishly with the concepts and techniques.

THANK YOU.

IT IS MY HONOR TO BE HERE.

Object-Oriented Programming Using PL/SQL

Professional help and support:
Kalyan Sircar
Juan Ortega
SOUG members, CIBER

Special THANK YOU!

Object-Oriented Programming Using PL/SQL

Contributors to presentation
1) **Kalyan Sircar** – support materials

2) **Allan Commander** – challenge question:
“If you want to object-oriented development, why wouldn't you just use Java? Oracle supports Java natively in the database!!”

3) **Dan McGhan** – Oracle OOP implementation critic:
“What a pain doing object-oriented development in PL/SQL is. Can't you just use packages and get the same benefits of code-reuse and library-style organization?

I'll be doing my best in the course of this presentation to answer Allan's and Dan's questions/challenges and at the end I hope they will agree that I did a good job.

That's one of my goals tonight.

I lied.

Troy lied (but it's OK, it's only natural).

I am not Mike Kemp. I am soug200910221900mike6543a&x*()yb.

It's nice to meet you.

I hope you like my instantiation.

BIG STATEMENT: Just as I can be represented (i.e., instantiated, go out of scope, be reinstantiated later with different attributes and methods) so can everything in the [***].

I lied.

Troy lied (but it's OK, it's only natural).

I am not Mike Kemp. I am soug200910221900mike6543a&x*()yb.

It's nice to meet you.

I hope you like my instantiation.

BIG STATEMENT: Just as I can be represented (i.e., instantiated, go out of scope, be reinstantiated later with different attributes and methods) so can everything in the UNIVERSE.

Object-Oriented Programming Using PL/SQL

OOP potential metaphors

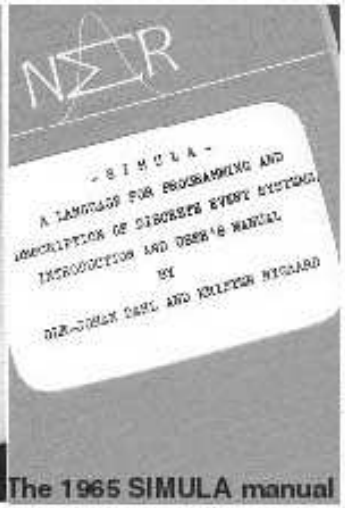
1) The Universe
2) Stars
3) Taxonomical heirarchy itself, or any node or end-point therein
4) Subatomic particles
5) Mortgage transaction
6) Mortgage
7) An annuity
8) A chemical reaction
9) A pinball machine
10) A pinball

What's the point?

Object-oriented development was implemented because it's implementers (pictures coming up!) thought this was the most direct, natural and accurate way of representing the world around us ..... via computer code.

Object-Oriented Programming Using PL/SQL

Ole-Johan Dahl (1978)



The 1965 SIMULA manual
The first object-oriented report

Kristen Nygaard (1978)

Ole-Johan Dahl
Kristen Nygaard

==========

Oslo, Norway – Norwegian Computing Center
Language: SIMULA
1962-1967
SIMULA never widely accepted.
SIMULA primarily used a Discrete Event Simulator.

Object-Oriented Programming Using PL/SQL

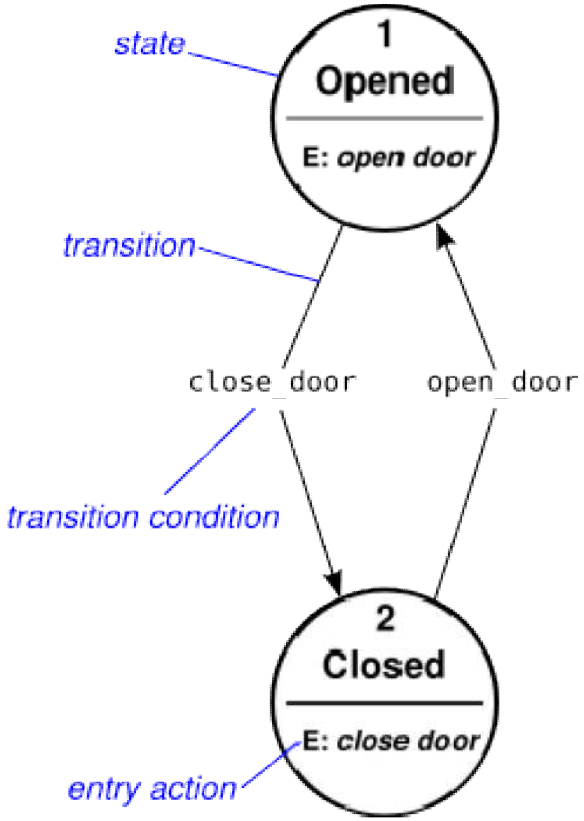


Diagram of a finite state machine.

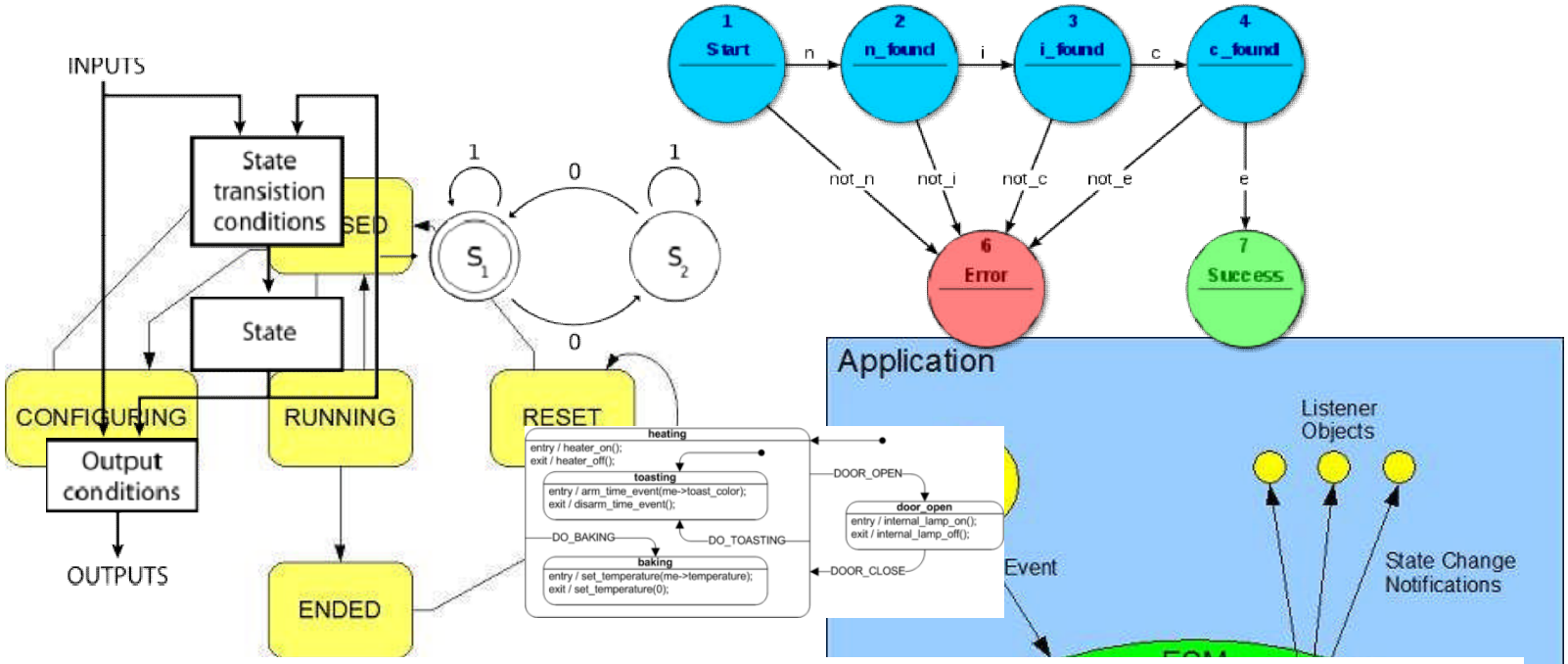
AKA software factory.

AKA state machine.

The “object” passes through these machines, and is altered by them.

The “obect” passes through these machines and *alters* them.

Object-Oriented Programming Using PL/SQL



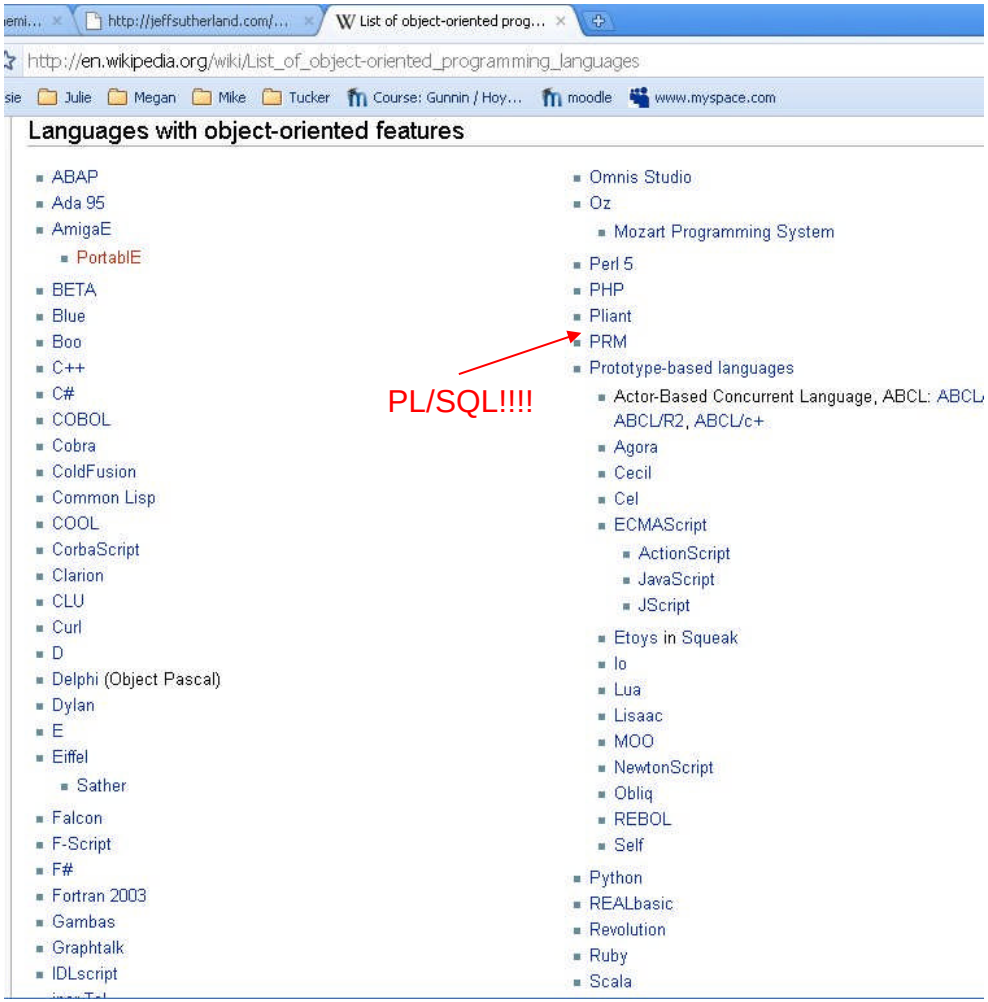
There is no limit to the way a “state machine” can be graphically represented, nor is there a limit to what physical object, process or conglomeration of any **combinations of objects and processes** can be represented by a “state machine”. In the end, a “state machine” is simply a **graphical representation of conditions and states.** Flowcharts do the same thing though the idea of a state machine extends out from a flowchart to create a richer graphical representation of more complex conditions and states.

Object-Oriented Programming Using PL/SQL

Over the course of several decades since the 1960's a multitude of programming languages with object-oriented capabilities (encapsulation, abstraction, inheritance, polymorphism, etc.) have been developed and come to prominence.

Right now, of course, the most prominent would likely be considered to be Java and C++.

Object-Oriented Programming Using PL/SQL



Object-Oriented Programming Using PL/SQL

Category	Ratings Oct 2009	Delta Oct 2008
Object-Oriented Languages	53.3%	-3.7%
Procedural Languages	42.5%	+3.0%
Functional Languages	2.7%	+0.4%
Logical Languages	1.4%	+0.2%

Statistical dominance of use of software languages by type.
NOTE: This indicates the use of the language not the implementation.
Source: www.tiobe.com

Object-Oriented Programming Using PL/SQL

Sample Java class

```
Java
  public class JavaClassExample{
    private String name;
    public void setName(String n){
      //set passed parameter as name
      name = n;
    }
     public String getName(){
      //return the set name
      return name;
    }
    public static void main(String args[]){
      JavaClassExample javaClassExample = new JavaClassExample();
      javaClassExample.setName("Visitor");
      System.out.println("Hello " + javaClassExample.getName());
    }
  }

  /*
    OUTPUT of the above given Java Class Example would be :
    Hello Visitor
  */
```

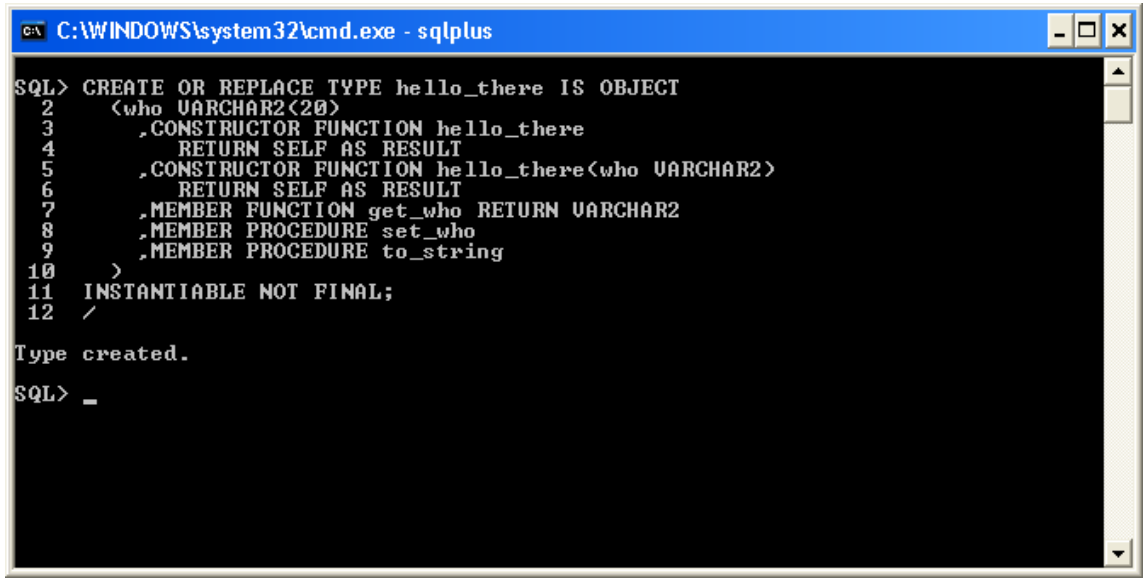
Object-Oriented Programming Using PL/SQL

Sample C++ class

```
#include &ltiostream.h>
class prime
{
   int      prime_cnt;
   int      prime_number;
 public :
    prime ( int num_in = 1, int prime_in = 2 )
      {
         prime_cnt     = num_in;
         prime_number  = prime_in;
     }
    prime& operator++();
    int get_nth_prime()
       {
         return ( prime_cnt );
       }
    int get_prime()
        {
          return ( prime_number );
        }
};
```

Object-Oriented Programming Using PL/SQL

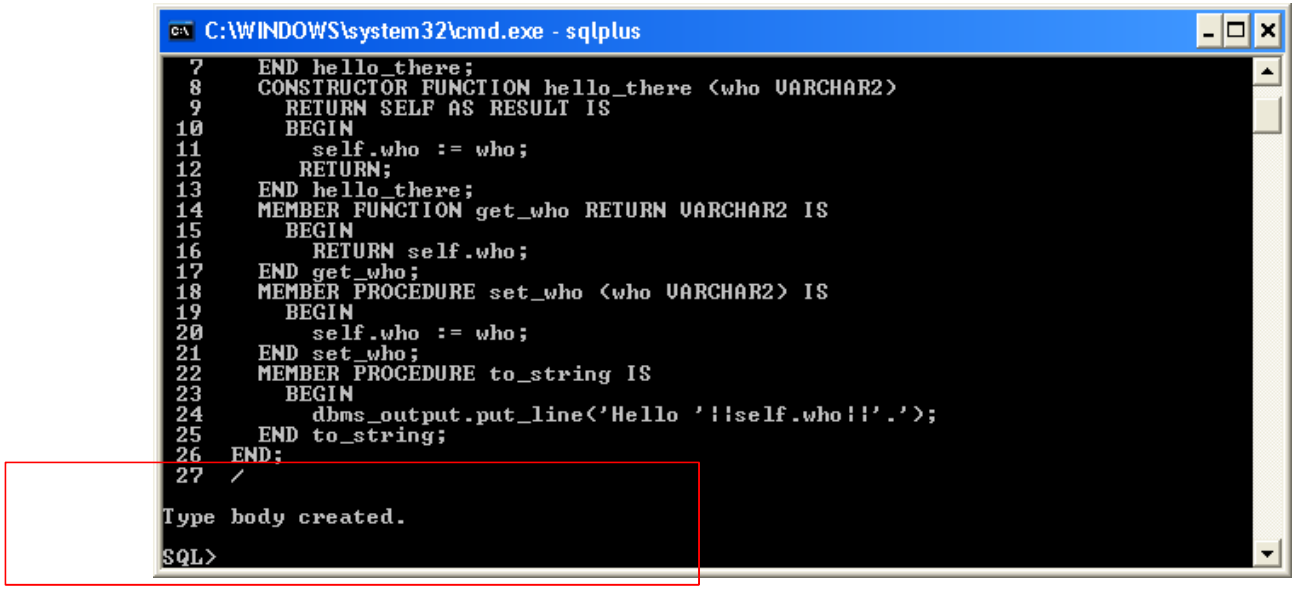
```
CREATE OR REPLACE TYPE hello_there IS OBJECT
  (who VARCHAR2(20)
    ,CONSTRUCTOR FUNCTION hello_there
       RETURN SELF AS RESULT
    ,CONSTRUCTOR FUNCTION hello_there(who VARCHAR2)
       RETURN SELF AS RESULT
    ,MEMBER FUNCTION get_who RETURN VARCHAR2
    ,MEMBER PROCEDURE set_who(who VARCHAR2)
    ,MEMBER PROCEDURE to_string
  )
INSTANTIABLE NOT FINAL;
/
```



Object-Oriented Programming Using PL/SQL

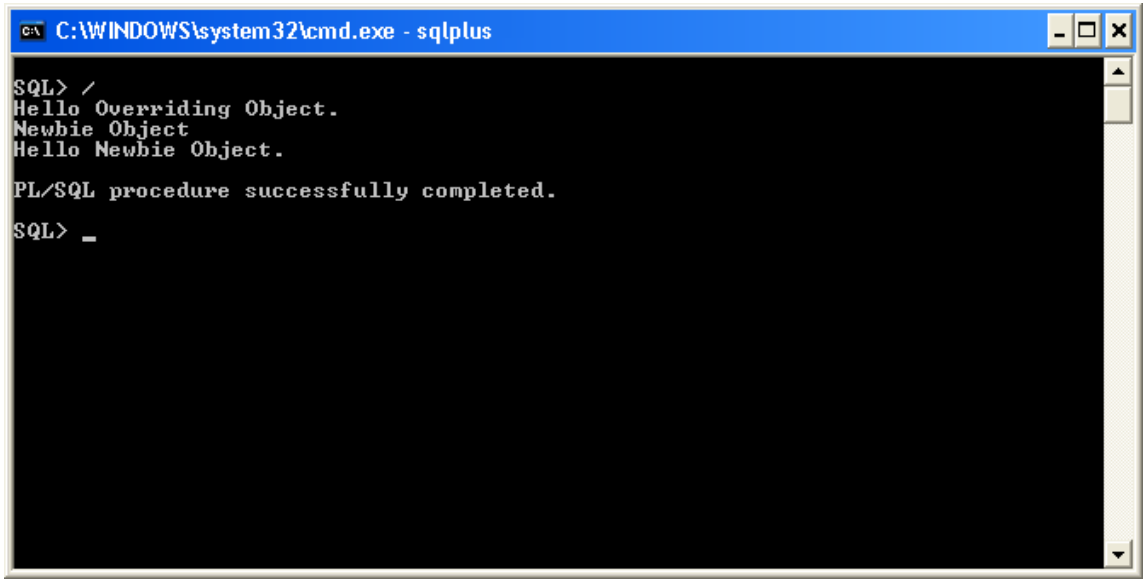
```
CREATE OR REPLACE TYPE BODY hello_there IS
  CONSTRUCTOR FUNCTION hello_there RETURN SELF AS RESULT IS
    hello HELLO_THERE := hello_there('Generic Object');
    BEGIN
      self := hello;
      RETURN;
  END hello_there;
  CONSTRUCTOR FUNCTION hello_there (who VARCHAR2)
    RETURN SELF AS RESULT IS
    BEGIN
      self.who := who;
     RETURN;
  END hello_there;
  MEMBER FUNCTION get_who RETURN VARCHAR2 IS
    BEGIN
      RETURN self.who;
  END get_who;
  MEMBER PROCEDURE set_who (who VARCHAR2) IS
    BEGIN
      self.who := who;
  END set_who;
  MEMBER PROCEDURE to_string IS
    BEGIN
      dbms_output.put_line('Hello '||self.who||'.');
  END to_string;
END;
```

Object-Oriented Programming Using PL/SQL



Object-Oriented Programming Using PL/SQL

```
DECLARE
  hello HELLO_THERE := hello_there('Overriding Object');
BEGIN
  hello.to_string();
  hello.set_who('Newbie Object');
  dbms_output.put_line(hello.get_who);
  hello.to_string();
END;
/
```



Object-Oriented Programming Using PL/SQL

Discussed Tonight

1) Origins and history of Object-Oriented Programming.
2) Theory behing the OOP concept.
2) Sample implementations of OOP principles in Java and C++.
3) Sample implementation of OOP principles in PL/SQL.

Available topics for future talks on OOP

1) STATIC MEMBER methods in PL/SQL.
2) How to compare object in PL/SQL (specialized functions MAP and ORDER).
3) What is inheritance and what can it do for me?
4) What is polymorphism and what can it do for me?
5) How to declare subclasses (A HELLO_THERE SUBCLASS THAT not only obtains your name, but matches your name with your preferred development language and displays both.)
6) TYPE EVOLUTION (Dan McGhan's contributing question!!)

SUMMARY of topics available for future talks: No history, no origins, no theory. NUTS AN BOLTS only!!

But for now ... to finish up ...

Object-Oriented Programming Using PL/SQL

First an anti-quote from Mr. Feurenstein:

I have never gotten trained up in object-oriented thinking. I spent about a month working with Java intensively, a few years ago, and all it confirmed for me was that, yes, it was possible to use an OO language like a procedural language."

So, Allan, so far it looks like you and the world's greatest PL/SQL Developer might be on the same page. Congratulations!! So far...

Object-Oriented Programming Using PL/SQL

Steven Feurenstein again:

From that starting point, Oracle has seen PL/SQL as a critical layer of technology in its stack, and committed substantial resources to improving the language. I had a scare around the time of the release of Oracle8i, when Oracle announced the ability to write Java Stored Procedures in the database. Would Oracle, as the rumors claimed, jettison PL/SQL in favor of Java? Ha! That would have been one heck of a mistake and the moment (and panicky feelings) passed quickly. PL/SQL is here to stay, Oracle continues to invest substantial resources in making it faster, easier, better.

So, from this, I take it that Oracle is seriously invested in making the PL/SQL Development language the best is can possibly be ... so “an” answer may be appearing through the mist.

Object-Oriented Programming Using PL/SQL

Michael McLaughlin (proud of the fact that he still has his hair when Steven doesn't) responds to the question this way:

Spatial, Oracle Text, and XML technologies all support the idea of an object-oriented development capability.

So, it appears to me that the answer to the challenge question that I CAN answer tonight appears to be twofold:

1) Oracle is seriously invested in making PL/SQL the best it can be.
2) Increasing requirements for access to complex datatypes make the change both relevant and timely.