1.1

* 1. Procedural constructs
  2. PL/SQL
  3. PL/SQL
  4. SQL
  5. PL/SQL
  6. SQL
  7. PL/SQL
  8. SQL

1. Allows for logic and control to be put together with basic sql statements to make programs with more uses.
2. Variables, constants, conditions, loops, and reusable code.
3. 1. Variables: v\_first\_name, v\_last\_name, v\_first\_letter
   2. Conditions: IF, ELSE, END IF
   3. Sql statement: SELECT … WHERE student\_id = 105.

1.2

* 1. Portability
  2. Blocks
  3. Exception

1. The app becomes more efficient when made in PL/SQL blocks allowing for less network traffic allowing for a faster app.
2. Improved performance, exception handling, and allowing integration of procedural constructs with SQL
3. PL/SQL requires either a database or Oracle tool, it’s object oriented programming techniques are not as extensive as C or Java, it’s the most efficient programming language to use with databases.
4. Create apps, manage app data, and manage a Oracle database.

1.3

1. 1. Anonymous PL/SQL block
   2. Functions
   3. Subprograms
   4. Compiler
   5. Procedures
2. 1. Optional: variables, user exceptions and cursors
   2. Mandatory: SQL or PL/SQL statements
   3. Optional: Actions to do when error happens
   4. Mandatory: End; (has to be with semicolon ender
3. 1. Fails: needs executable section to have at least 1 statement
   2. Fails: begin is missing.
   3. Fails: no statement in executable section
   4. Succeeds
4. 1. PL/SQL blocks that have no names are called anonymous blocks
   2. **Procedures** and **Functions** are named blocks and are stored in the database

BEGIN

dbms\_output.put\_line('Hello World');

END;

A screenshot of a computer

Description automatically generated



DECLARE v\_timestamp DATE;

BEGIN

SELECT ADD\_MONTHS(SYSDATE,6) INTO v\_timestamp FROM DUAL;

DBMS\_OUTPUT.PUT\_LINE('In six months, the date will be:'||v\_timestamp);

END;

A screenshot of a computer program

Description automatically generated