**Work through the following materials this week, making sure that you can do the given activities.**

1. **Read Sections 10.1, 10.4.1 & 26.1.1–2:**
   1. **Name the three basic approaches to database programming.**1. Embedding database commands in a general-purpose programming language. Embedded SQL.  
      2. Using a library of database functions or classes.   
      3. Designing a brand-new language.
   2. **Explain the concepts of:**
      1. ***impedance mismatch*** *is the term used to refer to the problems that occur because of differences between the database model and the programming language model.*
      2. ***stored procedures*** *procedures or functions—that are stored and executed by the DBMS at the database server*
      3. ***Triggers*** *Triggers are stored programs, which are automatically executed or fired when some events occur.*
2. **Read** [**Server-Side Programming: PL/SQL and Java.**](https://docs.oracle.com/cd/E11882_01/server.112/e40540/srvrside.htm#CNCPT036)**. Focus on server-side programming, PL/SQL and triggers; skip the section on “Overview of Java in Oracle Database”.**
   1. **Compare and contrast:**
      1. ***procedural* vs *non-procedural* languages.  
         NonProcedural:** the set of data to be operated on is specified, but not the operations to be performed or the manner in which they are to be carried out  
         **Procedural:** logic on the data. Loops and conditional branches.
      2. ***client-side* vs *server-side* database programming.  
         Client-Side:** use API or put SQL statements in source code. **Server-side:** can invoke stored subprograms. Can add triggers that fire on events.
   2. **Explain why one would want to do server-side programming using PL/SQL.**In order to use procedural programming on the database.
   3. **For each of the following code segments, identify the type of the database object and explain what it does.**
      1. **CREATE PROCEDURE limited\_count (limit IN integer) AS  
         BEGIN  
          FOR i IN 1..limit LOOP  
          dbms\_output.put\_line(i);  
          END LOOP;  
         END;**Procedure that prints out lines.
      2. **BEGIN  
          dbms\_output.put\_line('Hello, PL/SQL!');  
         END;**Anonymous block. Prints something out.
      3. **CREATE TRIGGER log\_trigger  
          BEFORE INSERT OR UPDATE OF lastName ON Person  
          FOR each row  
         BEGIN  
          dbms\_output.put\_line('Hello, name change!');  
         END;**

Is a trigger. Before a last name can be changed or inserted print out “Hello, name change!” for the rows that are changed.