Question 1 (pg 141):

Step 1: SET SERVEROUTPUT ON; Default is OFF.  
Step 2: BEGIN keyword informs the PL/SQL engine about the beginning of the PL/SQL block’s mandatory Execution section.  
Step 3: Display a message, using DBMS\_OUTPUT.PUT\_LINE(‘your message here’); in the execution section of the PL/SQL code, after the BEGIN keyword.  
Step 4: Finally, you must type “/” on a separate line to run the PL/SQL block.

Question 2 (pg 249):

Reusability of client logic: since the server side external program is shareable among all database users, the logic could be reused by connecting user.

Enhanced performance: moving the execution of calculative programs and methods from client to server side improves their execution by reducing the network round trips.

Logical extensibility: external routines maintain the margin to extend its logic. From an applications perspective, external procedures also avoid logical redundancy.

Integration of strengths: the realization of capabilities of a programming language in another one demonstrates flexibility of one and adaptation of another. In addition, a program adding up to the features of another language integrates the strengths and capabilities of programming.

Question 3 (pg 135):

Block structure is a set of SQL and PL/SQL statements that enable you to solve the specific problem. A PL/SQL block consists of three sections: the Declaration, Execution, and Exception section. Only the Execution section is required.

Procedural Language Capabilities: includes conditional logic (such as IF THEN ELSE), and iterative logic which includes WHILE, FOR, and basic LOOPs. These will run until the loop control becomes false. It also includes sequence logic, where statements are executed one after another.

Portability: you can run applications written in PL/SQL on any operating system and platform where the oracle database runs. With PL/SQL you can write portable programs and reuse them in different environments without any change at all.

Tight integration: PL/SQL allows you to use SQL commands, functions, and operators. You do not have to translate between SQL and PL/SQL data types. You can also construct SQL statements dynamically in PL/SQL.

Error Handling: Certain types of errors, called exceptions, can be trapped during the PL/SQL execution. Once an exception is trapped, corrective or diagnostic actions can be performed.

Question 4:

End User: this type of user retrieves data by initiating transactions or generating reports.

Database developer: this type of user designs and develops the interface. The applications can be custom made for a specific purpose, such as web-based applications, or it can be designed to be sold in mass quantities as a standalone product.

Database administrator: maintains a database management system. The DBA manages, monitors, tunes, and backs up the data and records in case of failure. The database management roles of the DBA include capacity planning, performance measurement, data recovery, data security, and user accounts management. In addition to this, the DBA deals with hardware and software upgrades.

Question 5:

Optional clauses:

1. Where: Limits the table horizontally, by only displaying records where a certain condition is met. For example, where a date is between a start and end date.
2. Group By: Used when aggregate functions have been used in the select statement. For example, you could group by product\_name where you have a SUM(price) in a sales table. This will display total sales value.
3. Having: A where clause for the group by. For example, records with a SUM(price) > 50000 will limit records to where the total sales are greater than 50000.
4. Order By: Orders the results in either ascending or descending order (based on a certain field).

Question 6:

Line 1: Incorrect syntax. Replace “setting serveroutput on” with “set serveroutput on;“.

Line 2: Incorrect syntax. Replace “declare variables” with “declare”.

Line 6: incorrect syntax. Replace “cursors info is” with “cursor info is”.

Line 14: Incorrect cursor name: Replace “for rec in information” with “for rec in info”

Line 15: Incorrect syntax. Replace “looping” with “loop”.

Line 18: Incorrect syntax (assignment operator). Replace “s\_qty = rec.quantity;” with “s\_qty := rec.quantity;”.

Line 24: Incorrect syntax. Replace “exit loop;” with “end loop;”.