```
   ID  DIVISION.
   PROGRAM-ID. Sample-Program-1.
   AUTHOR.  Husain Gholoom.
   ENVIRONMENT DIVISION.
   INPUT-OUTPUT SECTION.
   FILE-CONTROL.
      SELECT  Employee-File ASSIGN  TO
            'c:\HusainGholoom\cobol1-Programs\PG1.DAT'
            ORGANIZATION IS LINE SEQUENTIAL
            ACCESS MODE IS SEQUENTIAL.

      SELECT  Out-File ASSIGN  TO
            'c:\HusainGholoom\cobol1-Programs\PG1Out.DAT'
            ORGANIZATION IS LINE SEQUENTIAL
            ACCESS MODE IS SEQUENTIAL.
   DATA DIVISION.
   FILE SECTION.
   FD  Employee-File.
   01  Employee-Record.
      03  Employee-Name-IN  PIC X(19).
      03  Hours-Worked-IN   PIC 99.
      03  Rate-IN        PIC 99v99.

   FD  Out-File.
   01  Employee-Rec-Out     PIC X(80).

   WORKING-STORAGE SECTION.

   01  EOFILE-SWITCH     PIC  XXX VALUE 'NO'.
   01  WS-TOTAL         PIC  9(5)V99  VALUE  ZEROS.

   01  Header1.
      05    PIC  X(25)    VALUE  SPACES.
      05    pic  X(20)    VALUE 'Vodaphone Company'.

   01  Header2.
      05    PIC  X(5)    VALUE  SPACES.
      05    PIC  X(15)    VALUE 'Employee Name'.
      05    PIC  X(5)    VALUE  SPACES.
      05    PIC  X(15)    VALUE 'Hours Worked'.
      05    PIC  X(3)    VALUE  SPACES.
      05    PIC  X(8)    VALUE 'Rate'.
      05    PIC  X(5)    VALUE  SPACES.
      05    PIC  X(12)    VALUE 'Amount Paid'.

   01  Detail-Line.
      05    PIC  X(5)    VALUE  SPACES.
      05 Employee-Name-Out   PIC  X(19).
      05    PIC  X(5)    VALUE  SPACES.
      05 Hours-Worked-Out   PIC  9(2).
      05    PIC  X(10)    VALUE  SPACES.
      05 Rate-Out        PIC  99.99.
      05    PIC  X(10)    VALUE  SPACES.
      05 Amount-Paid-Out   PIC  9(5).99.
```

```
PROCEDURE DIVISION.


  OPEN INPUT   Employee-File
     OUTPUT   Out-File.

 DISPLAY HEADER1.
 DISPLAY HEADER2.

   READ   Employee-File
     AT END
        MOVE 'YES'  TO  EOFILE-SWITCH
     NOT  AT END
        MOVE  Employee-Name-IN  TO  Employee-Name-Out
        MOVE  Hours-Worked-IN   TO  Hours-Worked-Out
        Move  Rate-IN         TO  Rate-Out
        COMPUTE  WS-Total = Hours-Worked-IN * Rate-In
        Move  WS-Total       TO  Amount-Paid-Out
        DISPLAY  Detail-Line
   END-READ.

 CLOSE Employee-File   Out-File.


STOP RUN.
```

This code will be executed **ONLY ONCE** . The file still contains more records

---

```
  PERFORM  UNTIL EOFILE-SWITCH = 'YES'
     READ   Employee-File
       AT END
          MOVE 'YES'  TO  EOFILE-SWITCH
       NOT  AT END
          MOVE  Employee-Name-IN  TO  Employee-Name-Out
          MOVE  Hours-Worked-IN   TO  Hours-Worked-Out
          Move  Rate-IN         TO  Rate-Out
          COMPUTE  WS-Total = Hours-Worked-IN * Rate-In
          Move  WS-Total       TO  Amount-Paid-Out
          DISPLAY  Detail-Line
     END-READ

  END-PERFORM.
```

This code will be executed **Until End Of File is Reached**

```
PROCEDURE DIVISION.

100-MAIN-ROUTINE.

   PERFORM   200-OPEN-ROUTINE.

   PERFORM  300-HEADER-ROUTINE.

   PERFORM  UNTIL EOFILE-SWITCH = 'YES'
      READ   Employee-File
        AT END
           MOVE 'YES'  TO  EOFILE-SWITCH
        NOT  AT END
           PERFORM  400-CALCULATION-ROUTINE
      END-READ
   END-PERFORM.

   CLOSE  Employee-File   Out-File.

   STOP RUN.

200-OPEN-ROUTINE.
   OPEN INPUT   Employee-File
        OUTPUT  Out-File.

300-HEADER-ROUTINE.
   DISPLAY HEADER1.
   DISPLAY HEADER2.
```

Procedure division is divided into several blocks

```
   WRITE RECORD-OUT FROM Header1   AFTER PAGE.
   WRITE RECORD-OUT FROM Header2   AFTER 2.
```

```
400-CALCULATION-ROUTINE.

   MOVE Employee-Name-IN TO Employee-Name-Out .
   MOVE Hours-Worked-IN  TO Hours-Worked-Out .
   Move Rate-IN          TO Rate-Out  .
   COMPUTE  WS-Total = Hours-Worked-IN * Rate-In .
   Move WS-Total        TO Amount-Paid-Out .
   DISPLAY  Detail-Line  .
```

```
   WRITE RECORD-OUT FROM Header2   AFTER 2.
```