I. **Upcoming Test** – therefore we are not going onto the next module.
   - Next Tuesday we will cover Modules 10 and 11 out of the SAS textbook
     - One-Sample T-Test for $\mu$, $p$
     - Proc Means
     - Two-Sample T-Test
     - Proc Ttest

II. **Formatting Data** –
   - So far when dealing with data from the Utility data set we have read the date in as character and then created numeric values so that we could sort in month order instead of alphabetical order (of a character variable)
   - To change the format of a variable → can be done in 2 places
     1. In the INPUT line
        - We have done this in the formatted input
     2. In a FOMAT line
        - In the Data step you can use SAS defined formats (look in the help menu for a more complete list of formats)
        - **FORMAT <variable name> format name;**
          - **Date Formats**
            - MMDDYY6. → month day year (2 digits) → ex. (100599)
            - MMDDYY8. → month day year (4 digits) → ex. (10051999)
            - MMDDYY8. → month/day/year (2 digits) → ex. (10/05/99)
              - Anything can be used as the divider … ex. -, +, =, spaces, etc.
            - MMDDYY10. → month/day/year (4 digits) → ex. (10/05/1999)
            - MONYY6. → month (letters)/year (2digits)
            - MONDDYY9. → month($)/day/year → ex. (Oct/05/99)
          - **Dollar Formats**
            - DOLLAR8. → 8 wide, with 0 decimal places
            - DOLLAR8.2 → 8 wide, with 2 decimal places
          - **Exponential Formats**
            - E8.6 → 8 wide, with 6 decimal places (including the E)
        - Any combination of width (including a place for the decimal point) and the number of decimals assigned
        - The width must be at least 2 larger than the number of decimal places (1 for the number and 1 for the decimal point itself)
        - In exponential numbers the “E+02” must be accounted for in the decimal places (there must a minimum of 4 decimal places)
        - Dollar formats will not let you have an odd number of decimal places (either 2 or none)
        - Dollar signs and commas must be included in the width of your format.
EXAMPLE:

```
DATA EX1;
  INPUT dob mmddyy8. a b c;
CARDS;
  06/14/73 6354.948474764365 75.948555 746.24345
  8/27/77 874.948598949 653.7645 12.9099
  10/04/74 879988.0099039 7463.746353 8
;
PROC PRINT DATA=EX1;
RUN;
```

```
Obs   dob    a     b     c
  1  4913   6354.95 75.95  746.243
  2  6448   874.95  653.76 12.910
  3  5390  879988.01 7463.75  8.000
```

```
DATA EX2;
  INPUT dob mmddyy8. a b c;
  age = floor( (today() - dob)/364.25 );
CARDS;
  06/14/73 6354.948474764365 75.948555 746.24345
  8/27/77 874.948598949 653.7645 12.9099
  10/04/74 879988.0099039 7463.746353 8
;
PROC PRINT DATA=EX2;
RUN;
```

```
Obs   dob    a     b     c   age
  1  4913   6354.95 75.95  746.243  27
  2  6448   874.95  653.76 12.910  23
  3  5390  879988.01 7463.75  8.000  26
```

```
DATA EX2;
  INPUT dob mmddyy8. a b c;
  format a e10.5
         b 7.1
         c dollar7.2;
CARDS;
  06/14/73 6354.948474764365 75.948555 746.24345
  8/27/77 874.948598949 653.7645 12.9099
  10/04/74 879988.0099039 7463.746353 8
;
PROC PRINT DATA=EX2;
RUN;
```

```
Obs   dob    a    b    c         age
  1  4913  6.355E+03 75.9  $746.24
  2  6448  8.749E+02  653.8  $12.91
  3  5390  8.800E+05 7463.7  $8.00
```
Some Date Functions in SAS:

- `DATE()` → returns today’s Date as a SAS Date
- `DATEJUL(juliandate)` → returns a Julian date to a SAS date value
- `DATETIME()` → returns the current date and time of day
- `DAY(date)` → returns the day of the month from a SAS date value
- `JULDATE(date)` → returns the Julian date from a SAS date value
- `MDY(month, day, year)` → returns a SAS date value from the month day and year
- `MONTH(date)` → returns the month (as a number) from a SAS date
- `TIME()` → returns the current time of day
- `TODAY()` → returns today’s date as a SAS value
- `WEEKDAY(date)` → returns the day of week from a SAS date value
- `YEAR(date)` → returns the year from a SAS date value

Truncation Function:

- `CEIL(argument)` → smallest integer >= the argument
- `FLOOR(argument)` → largest integer <= the argument
- `INT(argument)` → the integer closest the argument
- `ROUND(argument, number of places)` → value to the nearest round off point

\[
\text{month} = \text{month}(	ext{dob})
\]