# Computer Science 1081 – Assignment #11

## Program #1

Write a program that uses a structure to pass around Movie Data information. Your program will have a function that prints all of the information about the structure, which will be called from your main function. Your main function will create two Movie Data variables, initializing them before calling the display function (one call per variable). One of your variables will be initialized using an initialization list, while the other will be initialized member by member. Please name the fields of your structure: title, director, year, and runtime. Please name your structure movieData, and the name of your display function displayMovie. Pass your structure to your function by constant reference parameter. You may use 2 movies of your choice to fill the data for the variables.

**Sample Outputs**:

```
Title       : War of the Worlds
Director    : Byron Haskin
Released    : 1953
Running Time: 88 minutes

Title       : War of the Worlds
Director    : Stephen Spielberg
Released    : 2005
Running Time: 118 minutes
Press any key to continue . . .
```

## Program #2

Write a program that stores information about 12 players in an array of structures. The structure (player) will contain 3 member fields: name, number, and points. When your program runs, it should call a function that will prompt the user to enter data for each of the fields for each of the 12 players. The function should take a reference parameter. Your program will then display a table of all of the players, and total points for the team. The number and name of the player who earned the most points should also be displayed.

**Sample Outputs**:

```
PLAYER
---------
Player's name: Ella
Player's number: 23
Points scored: 53
PLAYER
---------
Player's name: John
Player's number: 13
Points scored: 32
PLAYER
---------
Player's name: James
Player's number: 42
Points scored: 13
PLAYER
---------
Player's name: Frankie
Player's number: 82
Points scored: 14
PLAYER
---------
Player's name: Lindsey
Player's number: 84
Points scored: 24

PLAYER
---------
Player's name: Gabriel
Player's number: 46
Points scored: 24
PLAYER
---------
Player's name: Ruth
Player's number: 62
Points scored: 34
PLAYER
---------
Player's name: Fred
Player's number: 73
Points scored: 9
PLAYER
---------
Player's name: Molly
Player's number: 47
Points scored: 32
PLAYER
---------
Player's name: Nick
Player's number: 99
Points scored: 11
PLAYER
---------
```

```
Player's name: Karen
Player's number: 96
Points scored: 23
PLAYER
---------
Player's name: Terry
Player's number: 44
Points scored: 6



      NAME        NUMBER PTS SCRD
      Ella            23 53
      John            13 32
      James           42 13
      Frankie         82 14
      Lindsey         84 24
      Gabriel         46 24
      Ruth            62 34
      Fred            73 9
      Molly           47 32
      Nick            99 11
      Karen           96 23
      Terry           44 6

TOTAL POINTS: 275
The player who scored the most points is: Ella #23
Press any key to continue . . .
```

## Program #3

Write a program that calculates pay for either an hourly paid worker, or a salaried worker. Hourly workers will have a pay rate and hours worked, while salaried worker will have a salary and bonus. The program should create a structure that works for both of these types of workers.

The program should ask the user what type of worker they have information for, followed by questions related to the specific type of worker. You will create a function that will handle getting the input for the specific type of worker, using a reference parameter (getHourly & getSalaried). You will then call a print function that will use the information in the structure to print a report for that worker (call the function printWorker and use a const reference parameter).

The structure definition name should contain the word "worker". It will contain an enum, a double (for the gross pay), and a union of two sub-structures. The sub-structures will each have 2 fields, relating to the hourly and salaried data points. One sub-structure will be the hourly data; the other will be the salaried data.

I highly recommend the idea of nesting the structs and unions in a single definition. You can think of the outer struct as containing information that is common to all variations of the full structure, while also having a union for the features that are unique to each variation (those features possibly needing to be groups themselves into structures if the variation has more than one unique feature). The question you then have for yourself is how you tell which part of the union to use when interacting with the structure variable – the answer to that lies with the enum datatype, where you can define a structure member (in the outer structure) that is a datatype of an enum with values for each of the variations. You then set which variation your variable represents in the enum member, and utilize that variation's fields in the union. All while being able to reference these different variations with a single structure name.

**Sample Outputs**:
```
(H)ourly or (S)alaried? S
Enter the salary amount: 23058
Enter the bonus amount: 3802
Salaried Worker
Salary: $23058.00
Bonus: $3802.00
----------
Gross Pay: $26860.00
Press any key to continue . . .
====================================
(H)ourly or (S)alaried? H
Enter the number of hours worked: 39
Enter the hourly pay rate: 14.83
Hourly Worker
Hours: 39
Rate: $14.83
----------
Gross Pay: $578.37
```