Structures

C# Programming
Structures

©Rob Miles

Types

- We know that the *type* of a variable determines what you can put into it:
  - int can hold integers
  - float can hold real numbers
  - char can hold a single character
  - string can hold a string of text
- However, sometimes we want to hold slightly more complicated data

Cricket Data

- A cricket program would need to store the name of the player and their score:
  - string can hold the name
  - int can hold the score
- At the moment we have to use two arrays to hold this data for a number of players
- This is not a good idea, as we have to keep them in step all the time

A Player Structure

- C# lets us design the structure of a block of memory which can hold information about a player

```
struct Player {
    public string Name;
    public int Score;
}
```

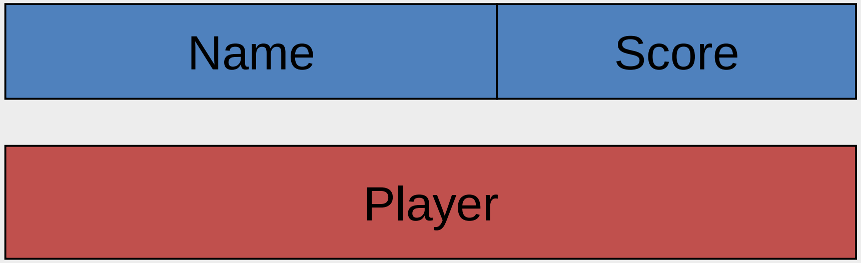
Using a Player

- Once we have our Player structure we can declare variables of that type:

```
Player p;
p.Name = "Fred";
p.Score = 99;
```

- We can also access the fields in the structure by using the notation shown above

Structures and Fields



Name

Score

Player

- A structure can contain any number of fields
- Each of them will be a particular type and have a particular identifier
- When a structure is created it is as a block of memory which holds the values for each field

Using structures

- You use structures just like you would any other variable

```
Player p, q;
p.Name = "Fred";
p.Score = 99;
q = p;
```

- In the final statement the values in `p` are copied into the values in `q`

Arrays of structures

- Since a structure is a type like any other you can create arrays of them:

```
Player [] team = new Player [11];
team[0].Name = "Fred";
team[0].Score = 20;
```

- Of course in a real program you would use a loop to go through an array

Structure Creation

- A structure is a design for a variable
- It is not a variable itself
- It must be created outside the Main method
- If you try to create it inside the method you will get loads of compilation errors

Structure and Design

- Structures are very useful from a design point of view
- Often you will come across items you need to store as a lump:
  - Customer record
  - Bank account
  - Game element
- Now you can just create a structure to hold the data in a single type

Structures in Method Calls

```
static void PrintPlayer(Player p)
{
 Console.WriteLine("Name:" + p.Name);
 Console.WriteLine("Score:" +
                             p.Score);
}
```

- You can pass the value of a structure into a method call just as you can any other variable type

Summary

- A structure is used to create a single object that contains a number of internal fields
- Once you have created a struct type you can then create variables of that type
- Structure variables can be used as any other variables in a program