DAILY ASSESSMENT FORMAT

Date:	**19/06/2020**	**Name:**	Nichenametla Bhargavi
Course:	C Programming	**USN:**	**4AL17EC061**
Topic:	**1: Structures & Unions** **2: Memory Management**	**Semester & Section:**	**6th Sem A Sec**
Github Repository:	Bhargavi_Nichenametla		

SESSION

Image of session

Report – Report can be typed or hand written for up to two pages.

Structure:

* A structure is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record.

* Defining a structure:
To define a structure, you must use the struct statement. The struct statement defines a new data type, with more than or equal to one member. The format of the struct statement is as follows:

```
 struct [structure name]
 {
        member definition;
        member definition;
...
        member definition;

 };
```

Union:

* A union is a special data type available in C that allows storing different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple purposes.

* Defining a Union:
To define a union, you must use the union statement in the same way as you did while defining a structure.
* The union statement defines a new data type with more than one member for your program.

The format of the union statement is as follows:

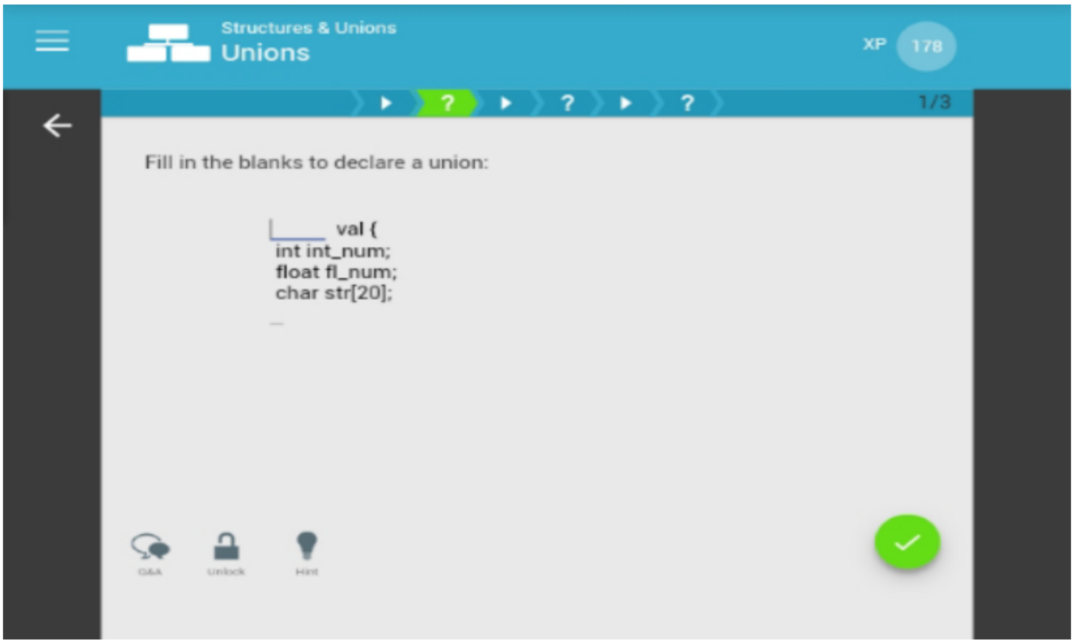
```
 union [union name]
 {
 member definition;
 member definition;
 ...
 member definition;
 };
```

Error Handling:

Error handling features are not supported by C programming, which is known as exception handling in C++ or in other OOP (Object Oriented Programming) languages. However, there are few methods and variables available in C's header file error.h that is used to locate errors using return values of the function call. In C, the function return NULL or -1 value in case of any error, and there is a global variable errno which sets the error code/number. Hence, the return value can be used to check error while programming.

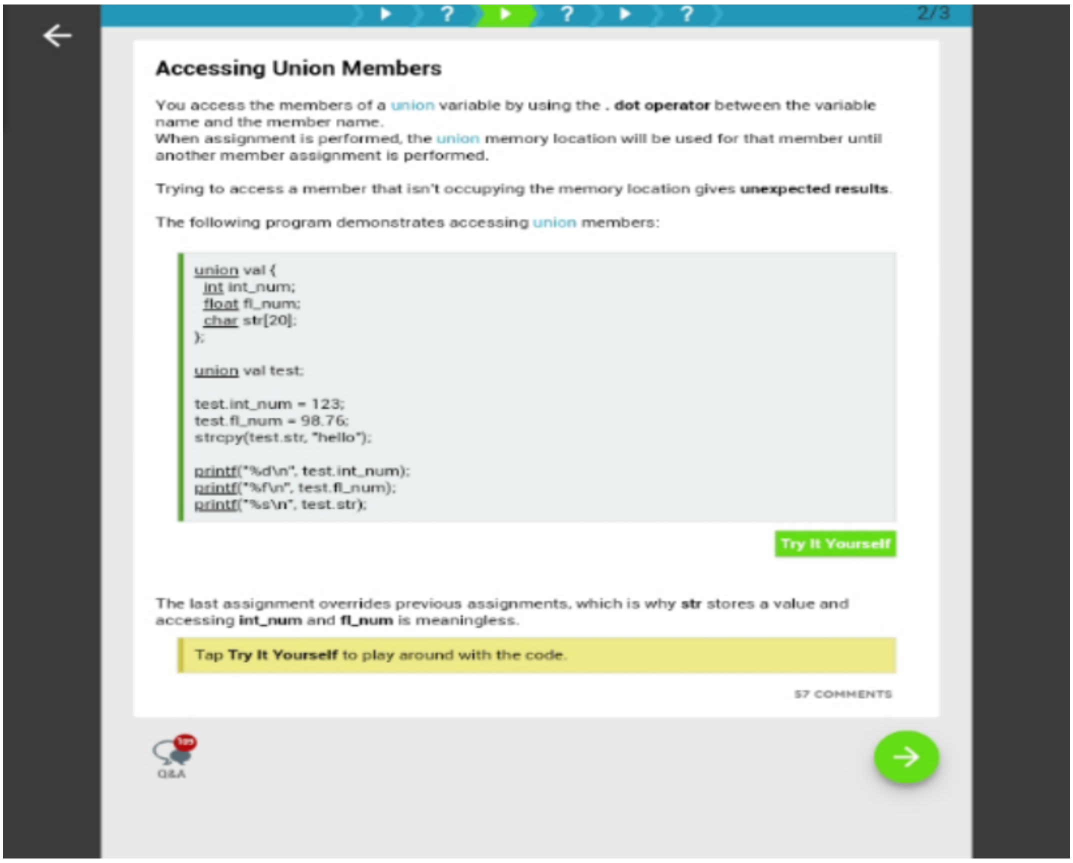
```
/* Divided By zero Error i.e. Exception*/#include <stdio.h>

#include <stdlib.h>
void main() {
 int ddend = 60;
 int dsor = 0;
 int q;
 if( dsor == 0){
 fprintf(stderr, "Division by zero! Exiting...\n");
 getch();
 exit(-1);
 }
 q = ddend / dsor;
 fprintf(stderr, "Value of quotient : %d\n", q);
 getch();
 exit(0);
}
```

Quiz was taken on each topic.



Fill in the blanks to declare a union:

```
____ val {
int int_num;
float fl_num;
char str[20];
_
```



Accessing Union Members

You access the members of a union variable by using the . **dot operator** between the variable name and the member name.
When assignment is performed, the union memory location will be used for that member until another member assignment is performed.

Trying to access a member that isn't occupying the memory location gives **unexpected results**.

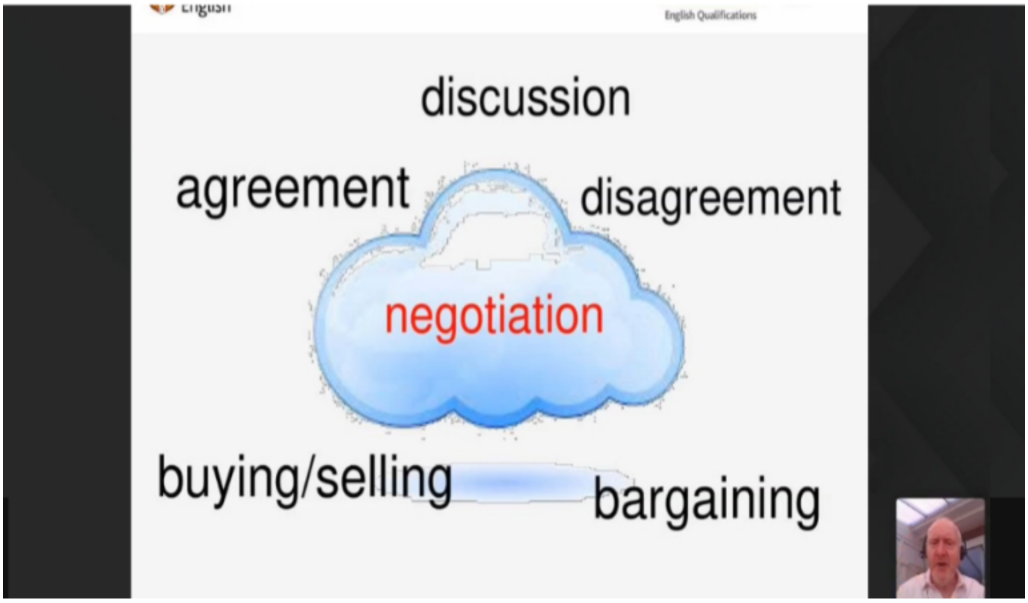
The following program demonstrates accessing union members:

```
union val {
  int int_num;
  float fl_num;
  char str[20];
};

union val test;

test.int_num = 123;
test.fl_num = 98.76;
strcpy(test.str, "hello");

printf("%d\n", test.int_num);
printf("%f\n", test.fl_num);
printf("%s\n", test.str);
```

The last assignment overrides previous assignments, which is why **str** stores a value and accessing **int_num** and **fl_num** is meaningless.

Tap **Try It Yourself** to play around with the code.

“Attended Webinar on “Skills for Work - Negotiation skills” conducted by Cambridge Assessment”



discussion

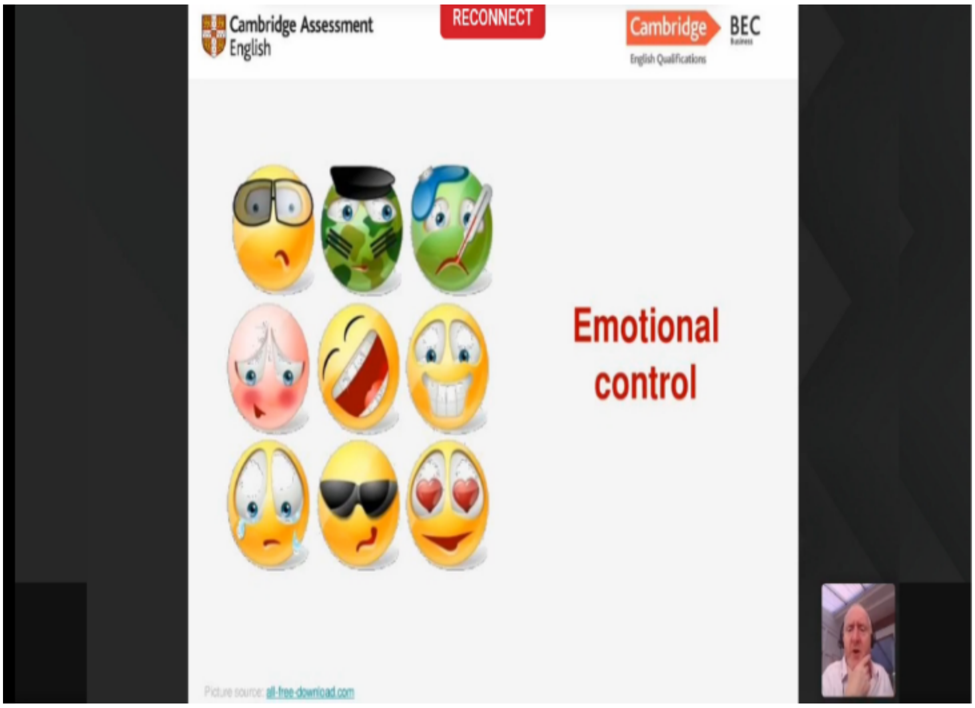
agreement

disagreement

negotiation

buying/selling

bargaining



Emotional control

Picture source: all-free-download.com