Content 37

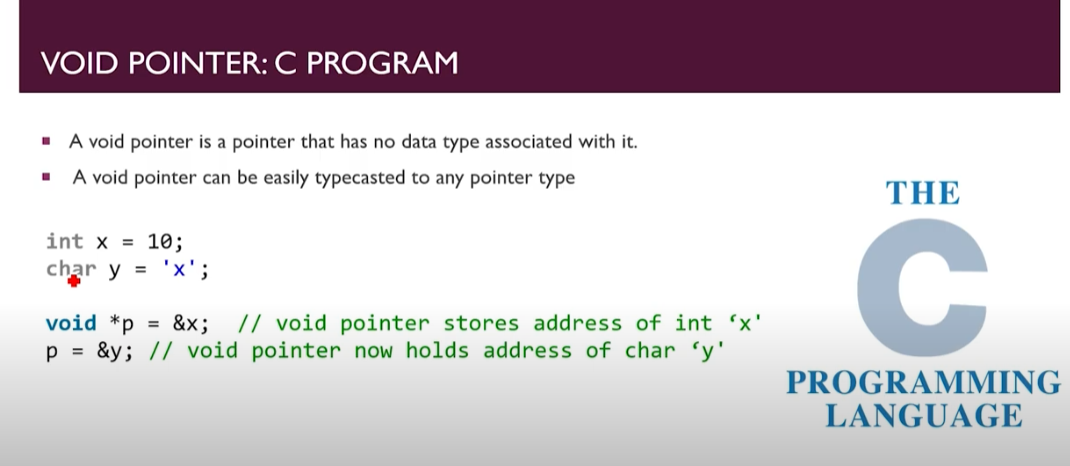
Void Pointer In C Language

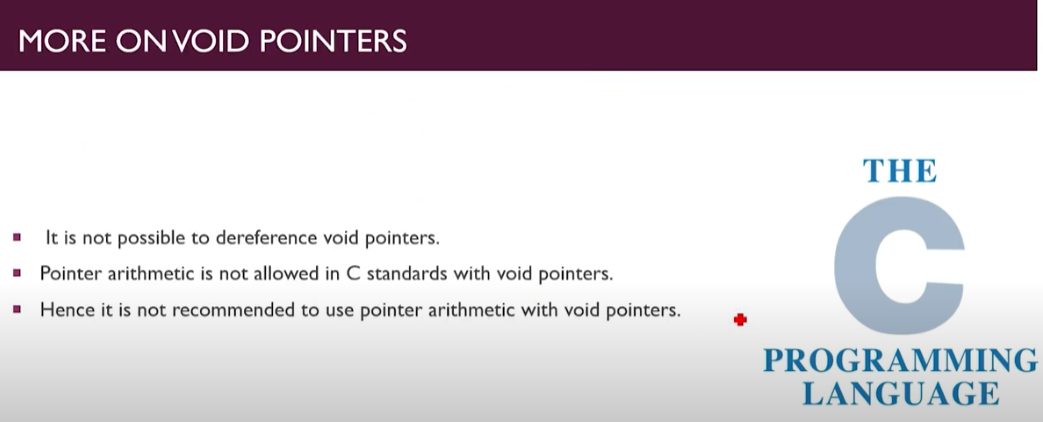
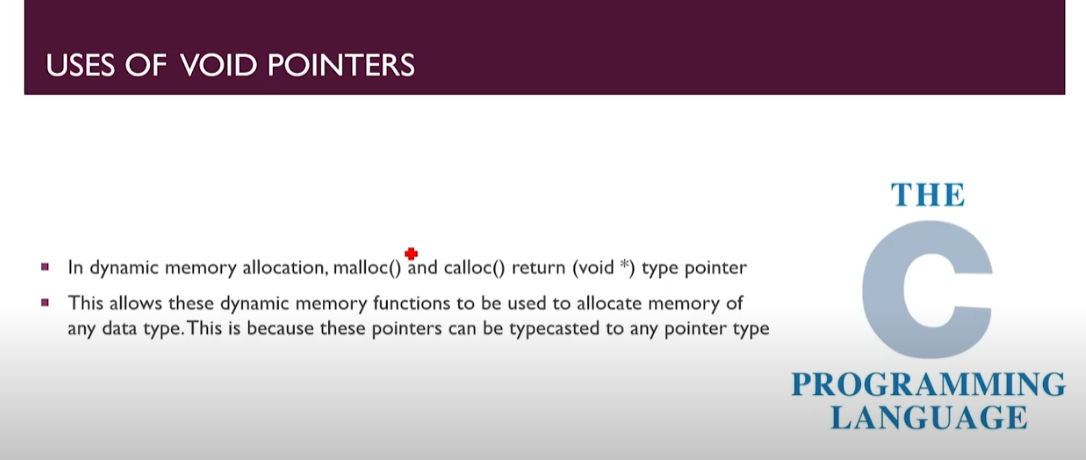
 we are going to learn about **Void pointers and their functionalities**. As we are already familiar, that void has no return type i.e., functions that are not returning anything are given the type void. So, in case of void pointers, they can be typecast into any data type whenever we want, meaning we do not have to decide a type for the pointer initially. In simple words, it is a general-purpose pointer variable.

* A Void Pointer is a data type with has no data type assigned with it.

**Example:-**  int \*ptr; = void \*ptr;

* A void pointer can be easily typecast at any pointer type later.
* In simple language it is a **General Purpose Pointer Variable.**

****



#### Important points:

* C does not allow void pointers to be dereferenced.
* We cannot use pointer arithmetic with void pointers.

Let us understand the dereference concept in a little bit more detail with the help of an example.

**Example 1:**

int a = 1;

void \* pointerrr;

pointerrr =&a;

printf("%d",\* pointerrr);

This program will, through a compile-time error, as we can not dereference a void pointer, meaning that we have to typecast the pointer every time it is being used.

Now let us take another example.

**Example 2:**

int a = 1;

void \* pointerrr;

pointerrr =&a;

printf("%d",\*(int\*) pointerrr);

In this example, the compiler will not throw any error and will directly output the result because we are using the type along with the pointer.

**So Typecasting is important here.**

**Code for Understanding Void Pointer:**

1. // Void Pointer Concept
2. #include <stdio.h>
3. int main()
4. {
5. int a = 26;
6. float b = 2.9;
7. void \*ptr;
8. ptr = &a;
9. // printf("The value of a is %d", \*ptr);// it will throw an error.
10. printf("The value of a is %d", \*((int \*)ptr));
11. ptr=&b;
12. printf("The value of b is %.2f", \*((float \*)ptr));
13. return 0;
14. }

**Output:**

The value of a is 26

The value of b is 2.90

But if we do like line no 9 it would throw error because it needs data type first.

**Code2:**

1. // Void Pointer Concept
2. #include <stdio.h>
3. int main()
4. {
5. int a = 26;
6. float b = 2.9;
7. void \*ptr;
8. ptr = &a;
9. ptr = &b;
10. // printf("The value of a is %d", \*ptr);    // it will throw an error.
11. printf("The value of a is %d\n", \*((int \*)ptr));
12. printf("The value of b is %.2f", \*((float \*)ptr));
13. return 0;
14. }

Here at line 8 pointer gets address of b (but not of a) and put that at line 11 and 12;

**Output:**

The value of a is 1077516698

The value of b is 2.90

So we get Garbage value at a.