Structure:

In C programming Struct (Structure) is collection of variables of different types under a single name.

How to declare a struct ?

struct structureName {

dataType member1;

dataType member2;

...

};

How to create struct varibales?

struct Person {

// code

};

int main() {

struct Person person1, person2, p[20];

return 0;

}

Another way of creating struct variable

struct Person {

// code

} person1, person2, p[20];

There are two types of operators used for accessing members of a structure.

. - Member operator

-> - Structure pointer operator

struct Person {

char name[50];

int citNo;

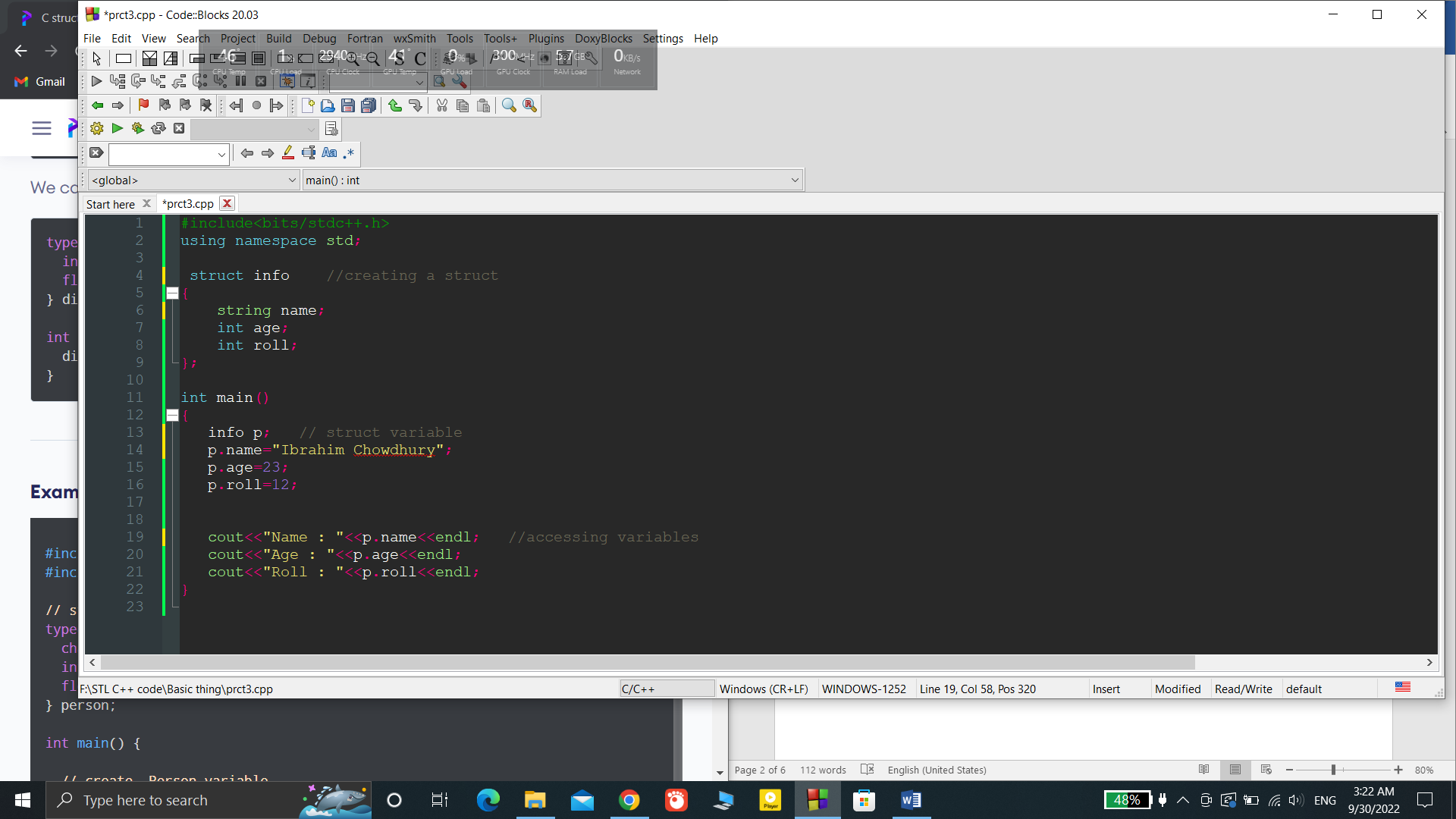
float salary;

}person1,person2;

If we want to access the member **salary** then we use then we use ,

person2.salary //accessing members of person2

code :



Keyword **typedef** :

Why we use typedef?

We use typedef keyword to create an alias name for data types to simplify the syntax of declaring variables.

typedef struct Distance {

int feet;

float inch;

} distances;

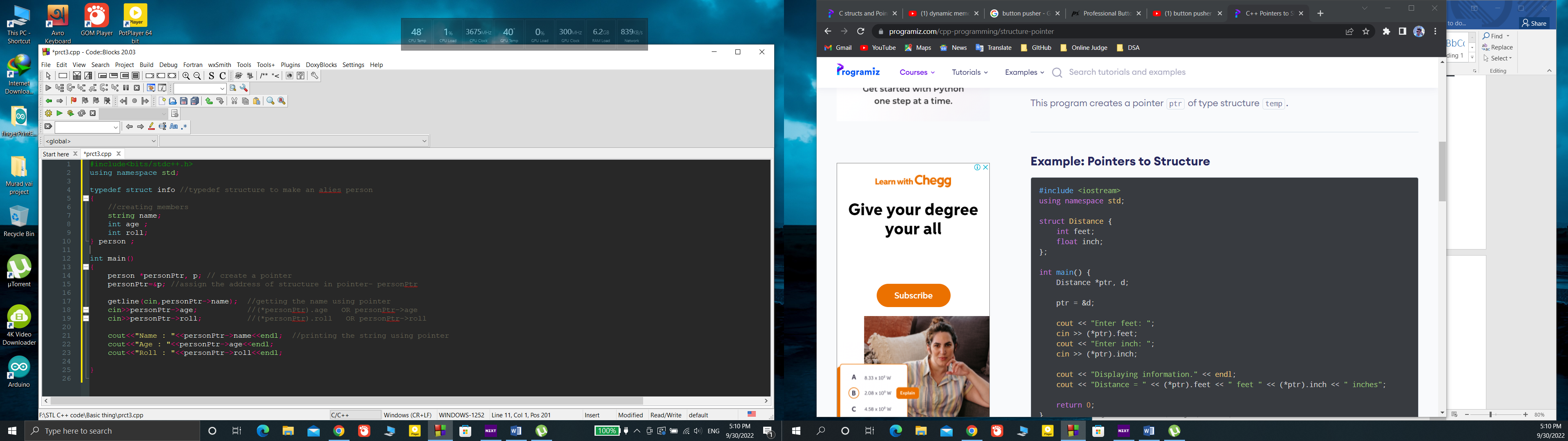
int main() {

distances d1, d2;

}

Here, we use typedef to make an alies **distances** of **Distance** struct type. Now we make variables of **distances** – **d1,d2**

Accessing members using pointers:



We use -> operation to accessing the member of the structure using pointer

**Dynamic Memory Allocation :**

malloc(),calloc(),realloc(),free()

**malloc()/ memory allocation:**

syntax:

ptr = (castType\*) malloc(size);

example :

ptr = (float\*) malloc(100 \* sizeof(float));

**calloc()/contegous allocation:**

the malloc function allocates memory and leaves the memory uninitialized but the calloc function allocates the memory and initialize all bits to zero.

Syntax:

ptr = (castType\*)calloc(n, size);

example:

ptr = (float\*) calloc(25, sizeof(float));

**free():**

syntax:

free(ptr);

**realloc():**

when the dynamically allocated memory is not sufficient then we can increase the size of the memory by using realloc() function.

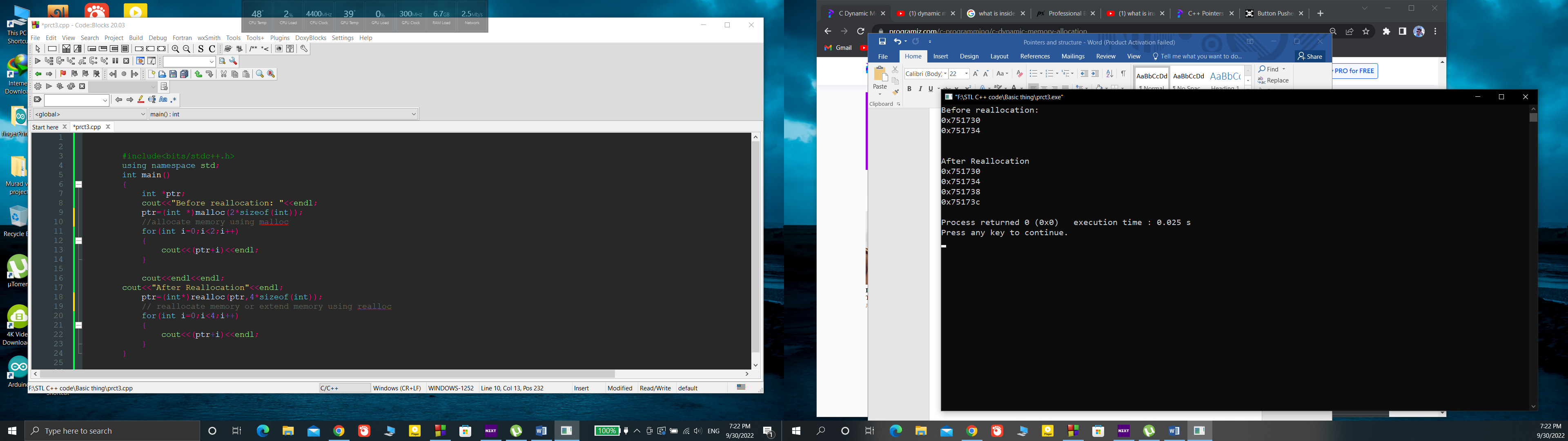
Syntax:

ptr = realloc(ptr, x);

example :

ptr = realloc(ptr, n2 \* sizeof(int));

example code:



Output:

