Using The "const" keyword

In C++, const is a keyword which is also called as **Qualifier.** Whenever we declare something as const then C++ compiler doesn't allow us to change its value once it has been intialized.
For example:
const float pi=3.14;
const int max_marks=100;
the above 2 declarations are creating pi and max_marks as **CONSTANT** and if we try to change their value the code will give syntax error.

There is another point to remember about const variables. The point is that whenever we declare a variable as const then it must be initialized along with declarations. So the following code will not even compile:

```
const float pi; //Syntax Error
pi=3.14;
```

Places Where "const" Can Be Used

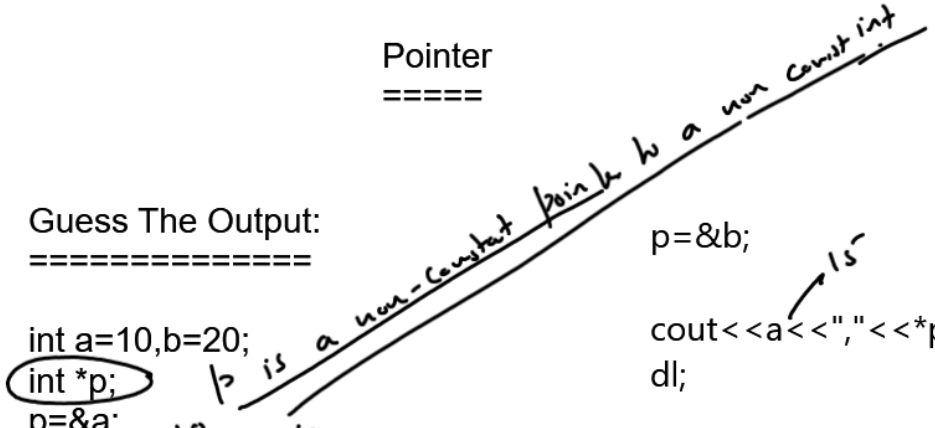
The keyword "const" in C++ can be used at 8 places and they are:

1. "const" variable
2. pointer to "const"
3. "const" pointer
4. "const" pointer to "const"
5. "const" function argument
6. "const" data member
7. "const" member function
8. "const" object

Pointer
=====

Guess The Output:
==========

```
int a=10,b=20;
int *p;
p=&a;
cout<<a<<","<<*p<<","<<b<<endl;

a=12;
cout<<a<<","<<*p<<","<<b<<endl;

*p=15;
cout<<a<<","<<*p<<","<<b<<endl;

p=&b;
cout<<a<<","<<*p<<","<<b<<endl;

b=25;
cout<<a<<","<<*p<<","<<b<<endl;

*p=30;
cout<<a<<","<<*p<<","<<b<<endl;
```



Observation of the previous code
==========

1. The pointer p can change the value of the variable to which it is pointing

2. The pointer p can point to another variable as and when required

3. This means **P IS A NON CONSTANT POINTER TO A NON CONSTANT INTEGER**

Pointer to "const"

A pointer to const is a pointer which is declared in such a way that it can access or read the value of the variable to which it is pointing but it cann't change that variables value.

The general syntax of declaring pointer const is :

const <data_type> *<ptr_name>;
or
<data_type> const *<ptr_name>;

Example:

const int *p;
or
int const *p;
The above declarations will be read as p is a non constant pointer to a constant integer.

.

Guess The Output:
==========

```
int a=10,b=20;
const int *p;
p=&a;

a=12;

*p=15;
```

```
p=&b;

b=25;

*p=30;
```

Constant pointer

A constant pointer is a pointer which cann't change the address assign to it.

It means that once we initialize this pointer to some address then it can not point to some other address.

Thus a constant pointer through out its life time points to only one variable.

Special point
==========
Reference variable and array name, both are internally constant pointer

Syntax
======
<data_type> * const <ptr_name>=<some address>;
Example:
=====
int a=10;
int * const p=&a; // p is a constant pointer to a non constant integer

Guess The Output:
==========

int a=10,b=20;
int *const p;// error
int * cont p=&a;
p=&a; ✗

a=12; ✓

*p=15; ✓

p=&b; ✗

b=25; ✓

*p=30; ✓