A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Pass by Value** | **Pass by Reference** | **Pass by Address** |
| Passes actual value of argument into parameters of function | Passes reference of an argument into parameters of function | Passes address of an argument into parameters of function |
| Ex: a=5 | Ex: &a | Ex: \*a |
| Changes made to the parameter inside the function have no effect on the argument | Changes made to the parameter inside the function affect the argument. | Changes made to the parameter inside the function affect the argument. |
| In C++, CbyValue is default.  Mean: code inside function cannot alter arguments used to call function | Address is used | Pointer is required |
| Ex: passbyvalue(a,b) | Ex: passbyref(a,b); | Passbyaddr(&a,&b); |
| void passbyvalue(int x, int y)  {  int z=x;  x=y;  y=z;  } | void passbyref(int &x, int &y)  {  int z=x;  x=y;  y=z;  } | void passbyaddr(int \*x, int \*y)  {  int z= \*x;  \*x=\*y;  \*y=z;  } |

CODE 1:

#include<iostream>

using namespace std;

void passbyvalue(int x, int y)

{

int z=x;

x=y;

y=z;

}

void passbyref(int &x, int &y)

{

int z=x;

x=y;

y=z;

}

void passbyaddr(int \*x, int \*y)

{

int z= \*x;

\*x=\*y;

\*y=z;

}

int main()

{

int a=5, b=6;

cout<<"Before Swapping"<<endl<<"a is "<<a<<endl<<"b is"<<b<<endl;

// passbyvalue(a,b);

// passbyref(a,b);

passbyaddr(&a,&b);

cout<<"After Swapping"<<endl<<"a is "<<a<<endl<<"b is"<<b<<endl;

return 0;

}

Code: 2

#include<iostream>

using namespace std;

void passbyval(int val)

{

val = 10;

cout<<"passed number is"<<val<<endl;

}

void passbyref(int &ref)

{

ref = 20;

cout<<"passed number is"<<ref<<endl;

}

void passbyaddr(int \*ptr)

{

\*ptr = 30;

cout<<"passed number is"<<\*ptr<<endl;

}

int main()

{

int x=5;

cout<<"x is"<<x<<endl;

passbyval(x);

cout<<"final val is "<<x<<endl;

cout<<"x is"<<x<<endl;

passbyref(x);

cout<<"final val(ref) is"<<x<<endl;

int \*xptr = &x;

cout<<"x is"<<x<<endl;

cout<<"xptr is "<<\*xptr<<endl;

passbyaddr(xptr);

cout<<"final val(ptr) is"<<\*xptr<<endl;

return 0;

}