REFERENCES, POINTERS
PASSING PARAMETERS TO FUNCTIONS

Problem Solving with Computers-I

```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```

Pass by value

```
 void swapValue(int x, int y){
     int tmp = x;
     x = y;
     y = tmp;
 }
int main() {

    int a=30, b=40;

     cout<<a<<" "<<b<<endl;

    swapValue( a,  b);

    cout<<a<<" "<<b<<endl;

}
```

What is printed by this code?

A.

```
30 40
30 40
```

B.

```
30 40
40 30
```

C. Something else

References in C++

A reference in C++ is an alias for another variable

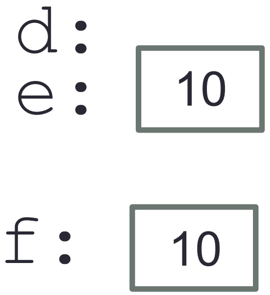
```
int main() {
  int d = 5;
  int &e = d;
}
```

References in C++

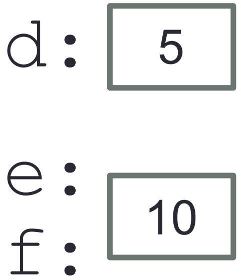
```
int main() {
  int d = 5;
  int &e = d;
  int f = 10;
  e = f;

}
```

How does the diagram change with this code?

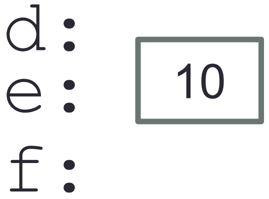
A.



B.



C.



D. Other or error

Passing parameters by reference

```
 void swapValue(int x, int y){
     int tmp = x;
     x = y;
     y = tmp;
 }
int main() {
     int a=30, b=40;
     swapValue( a,  b);
     cout<<a<<"  "<<b<<endl;
}
```

Pointers

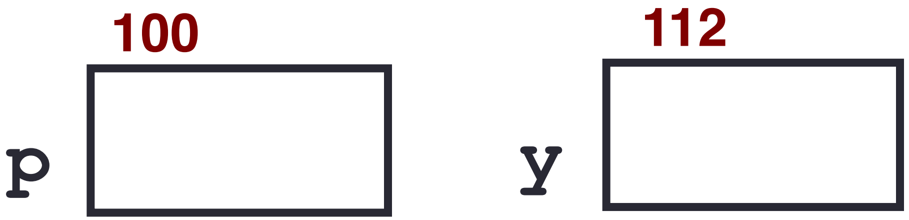
- Pointer: A variable that contains the address of another variable
- Declaration: *type ** pointer_name;

```
int* p;
```

MAN, I SUCK AT THIS GAME. CAN YOU GIVE ME A FEW POINTERS?
0x3A28213A 0x6339392C, 0x7363682E.
I HATE YOU.

How to make a pointer point to something

```
int *p;
int y = 3;
```

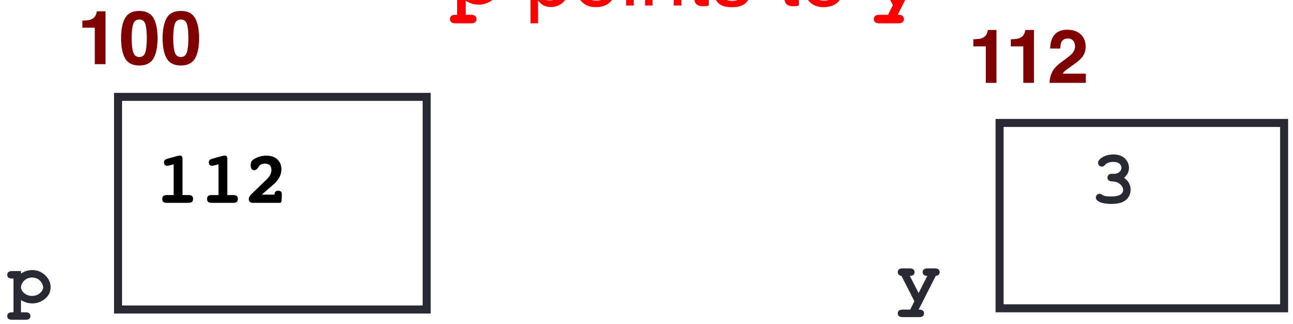


To access the location of a variable, use the address operator ‘&’

Pointer Diagrams:
Diagrams that show the relationship between pointers and pointees

p points to **y**

Pointer: p

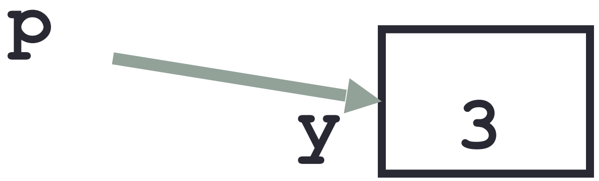


Pointee: y

You can change the value of a variable using a pointer !

```
int *p, y;
y = 3;
p = &y;

*p = 5;
```

Two ways of changing the value of a variable



- Change the value of y directly:

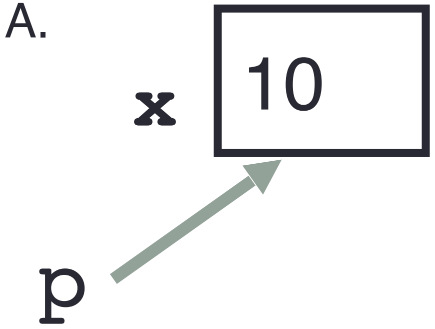
- Change the value of y indirectly (via pointer p):

Tracing code involving pointers

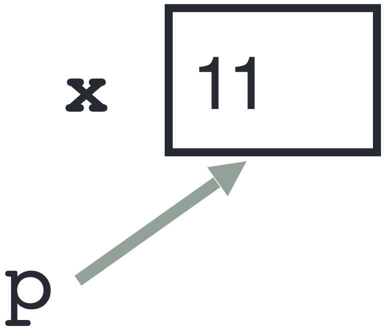
```
int *p;
int x=10;
p = &x;
*p = *p + 1;
```

Q: Which of the following pointer diagrams best represents the outcome of the above code?

A.



B.



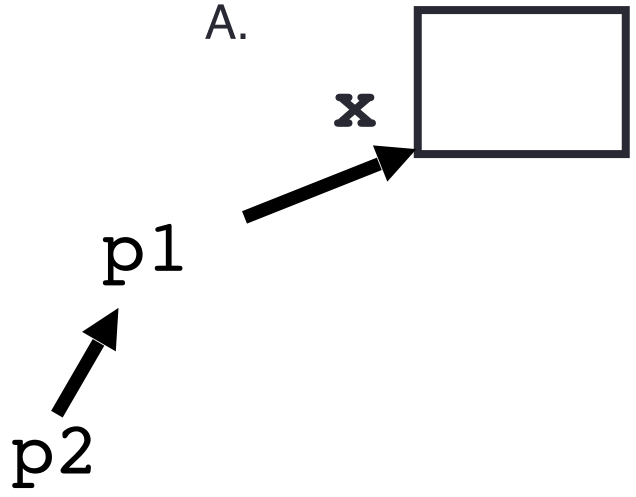
C. Neither, the code is incorrect

Pointer assignment

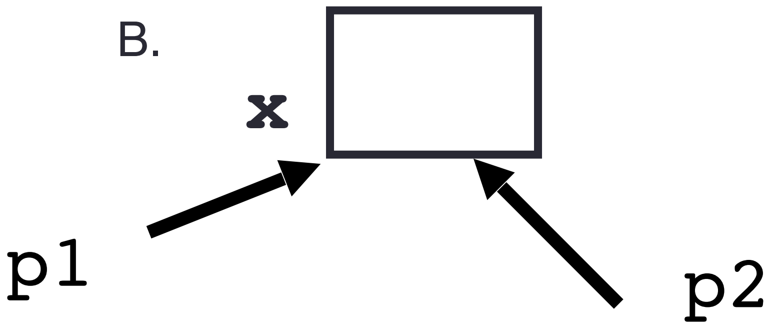
```
int *p1, *p2, x;
p1 = &x;
p2 = p1;
```

Q: Which of the following pointer diagrams best represents the outcome of the above code?

A.



B.



C. Neither, the code is incorrect

Passing parameters by address

```
 void swapValue(int x, int y){
     int tmp = x;
     x = y;
     y = tmp;
 }
int main() {
    int a=30, b=40;
    swapValue( a,  b);
    cout<<a<<"  "<<b<<endl;
}
```

Next time

- Arrays and pointers
- Structs