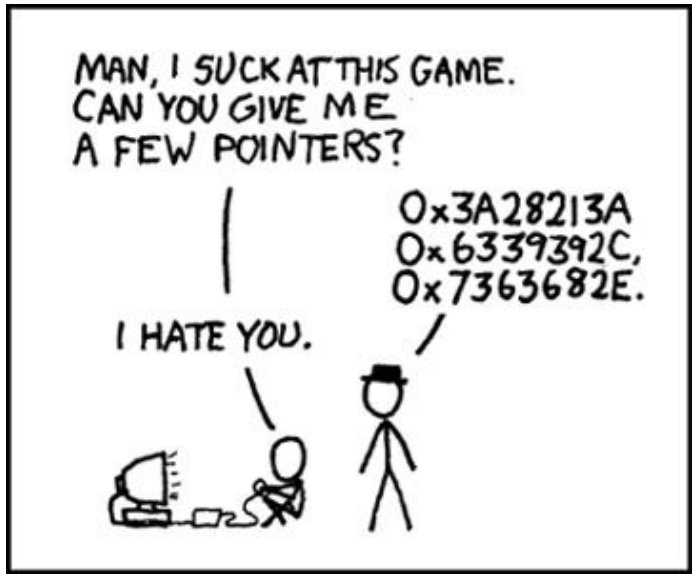
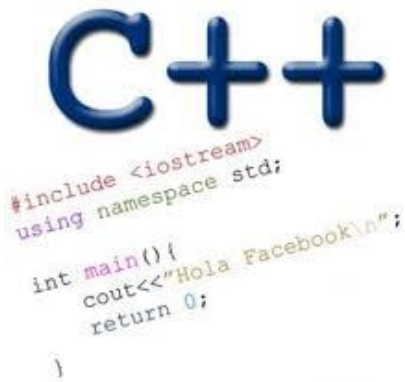
POINTERS



Problem Solving with Computers-I

https://ucsb-cs16-wi17.github.io/



How comfortable do you feel with using github?

A. Very comfortable in the context of labs, I have a basic understanding of how git works
B. I know how to use it but I have no idea how git works
C. I don’t feel comfortable using it
D. I am completely lost

How far along are you with lab04

A. Almost done
B. I am on track to finish
C. I am stuck and don’t know how to proceed
D. Haven’t started

Swap function – midterm 1

```
#include <iostream>
using namespace std;
void swap(int a, int b){
        cout<< "Inside swap"<<endl;
        int tmp = a;
        a = b;
        b = tmp;
        cout<< a << "  " << b<< endl;
}

int main(){
        int x= 10, y=20;
        cout<< "Before swap" <<endl;
        cout<< x<< "  " <<y<<endl;
        swap(x, y);
        cout<< "After swap" <<endl;
        cout<< x<< "  " <<y<<endl;
}
```

Pointers

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

```
int *p;
```

How do we initialize a pointer?

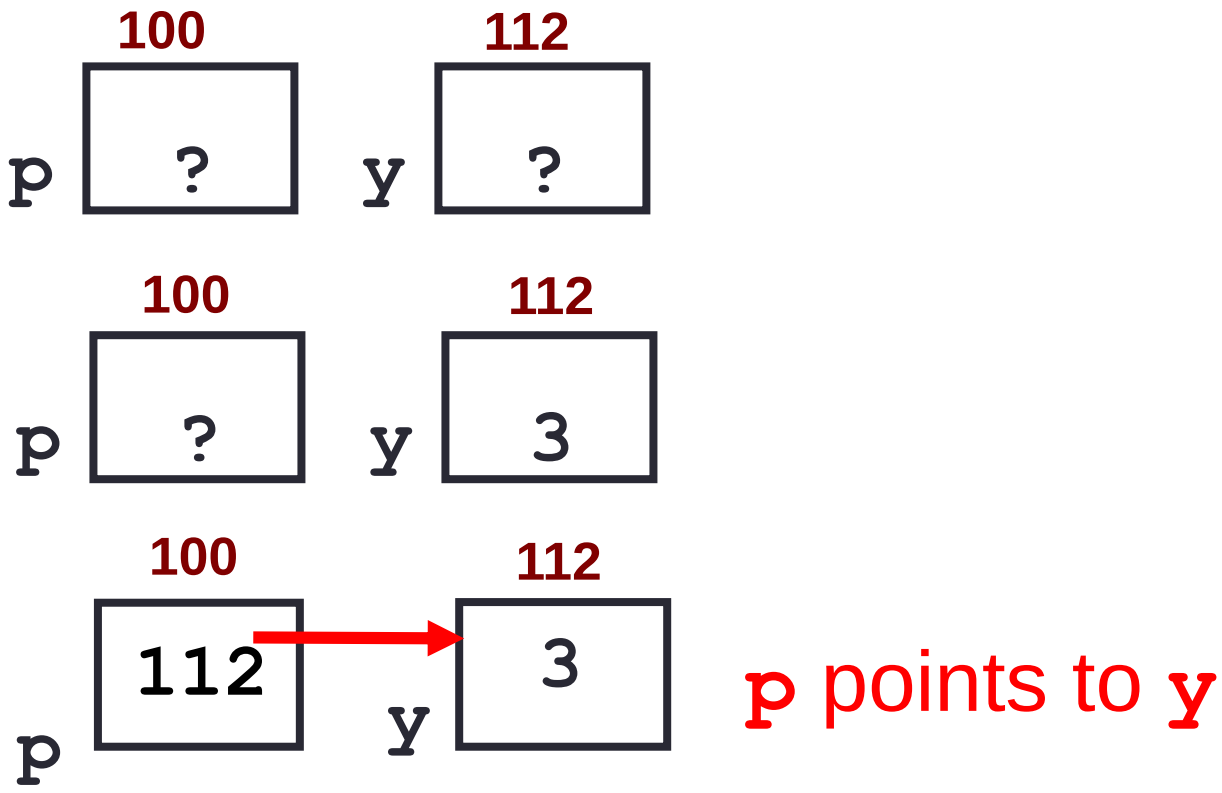
How to make a pointer point to something

```
int *p;
int y;
```

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

How to make a pointer point to something

```
int *p, y;
```



p points to **y**

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

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

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

*p = 5;
```

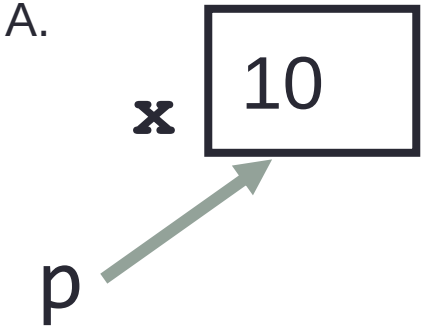
Use dereference * operator to left of pointer name

Tracing code involving pointers

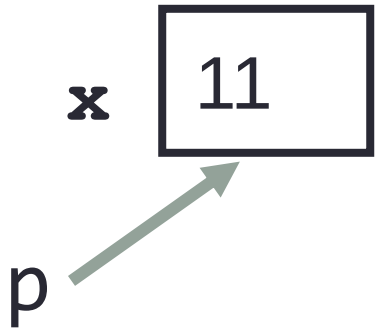
```
int *p, x=10;
p = &x;
*p = *p + 1;
```

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

A.

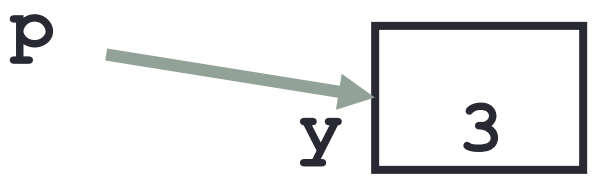


B.



C. Neither, the code is incorrect

Two ways of changing the value of a variable



Change the value of y directly:

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

Pointer assignment and pointer arithmetic: Trace the code

```
int x=10, y=20;

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

p2 = p1;

int **p3;

p3 = &p2;
```

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.

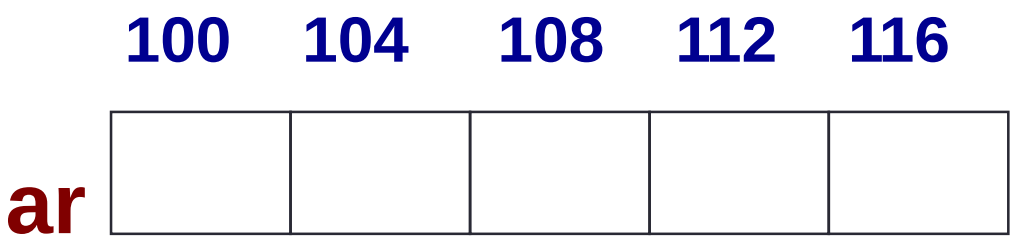
B.

C. Neither, the code is incorrect

Swap function

```
int main(){
        int x= 10, y=20;
        cout<< “Before swap” <<endl;
        cout<< x<< “  ” <<y<<endl;
        swap(x, y);
        cout<< “After swap” <<endl;
        cout<< x<< “  ” <<y<<endl;
}
```

Arrays and pointers



- `ar` holds the address of the first element (like a pointer)
- `ar[0]` is the same as `*ar`
- Use pointers to pass arrays in functions

```
int ar[5]={65, 85, 97, 75, 95};

int *p;
```

Next time

- What can go wrong when using pointers
- References
- Pointers and structs
- Mechanics of function calls contd.–call by reference