STRUCTS REFERENCES POINTERS (REVIEW)

Problem Solving with Computers-I

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

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

C++ structures (lab05)

A **struct** is a data structure composed of simpler data types.

```
struct Point {
    double x; //member variable of Point
    double y; //member variable of Point
};
```

Think of Point as a new data type

```
Point p1;              // Declare a variable of type Point
Point p1 = { 10, 20};  //Declare and initialize
```

C++ structures (lab05)

- A **struct** is a data structure composed of simpler data types.

```
struct Point {
     double x; //member variable of Point
     double y; //member variable of Point
};
```

- Access the member variables of p1 using the dot ‘.’ operator

```
Point p1;
p1.x = 5;
p1.~~x~~y = 10;
```

- Access via a pointer using the -> operator

```
Point* q = &p1;
(*q).x = 5;
(*q).~~x~~y = 10;
```

Which of the following is/are incorrect statement(s) in C++?

```
struct Point {
    double x;
    double y;
};
```

```
struct Box {
        Point ul; // upper left corr
        double width;
        double height;
};
```

[✓ A.] ul.x = 10; Incorrect — the ul member variable of Box may only be accessed via an object of Box

B. Box b1 = {{500, 800}, 10, 20}; → correct

C. Box b1, b2; b1.ul = {500, 800};

This works only with C++11

D. A and C

E. None of the above are incorrect

Box b1;

b1 is an object of Box

Passing structs to functions

- Write a function that prints the x and y coordinates of a `Point`
- Write a function that takes a `Point` as parameter and initializes its x and y coordinates

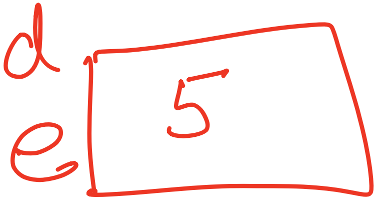
See code written in class

References in C++

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

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

int &e = d;

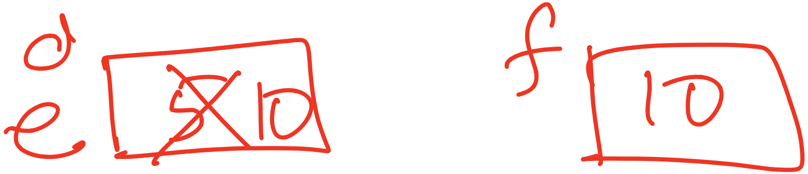


Changing d will change e and vice versa

e is a reference
'nick name' for d

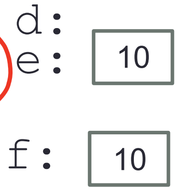
References in C++

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



How does the diagram change with this code?

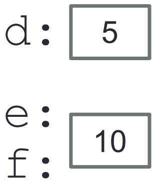
[✓ A.]



d:
e: 10

f: 10

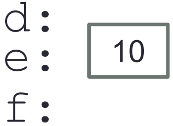
B.



d: 5

e:
f: 10

C.



d:
e: 10
f:

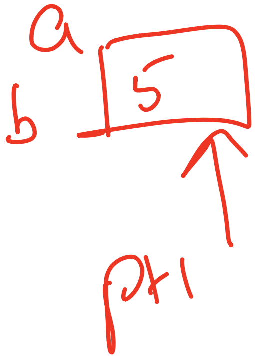
→ wrong

D. Other or error

f is a separate entity in memory.

Pointers and references: Draw the diagram for this code

```
int a = 5;
int & b = a;
int* pt1 = &a;
```

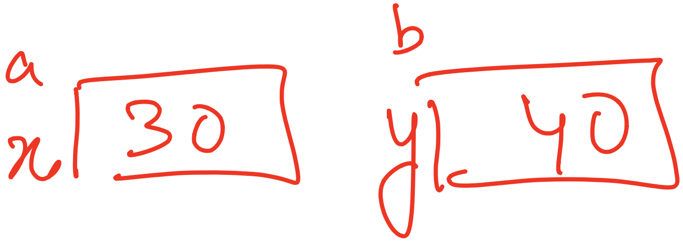


What are three ways to change the value of ‘a’ to 42?

Call by reference: Modify to correctly swap a and b

```
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;

}
```

x&y are references to a &b
swapping x&y will result in
swapping a&b.



Passing structs to functions by reference

- Write a function that takes a `Point` as parameter and initializes its x and y coordinates

This was done using pass by reference

Refer to code written in lecture.

Next time

- Arrays of structs
- Dynamic memory allocation