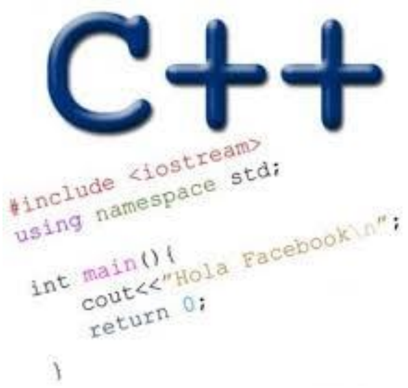
FRIEND FUNCTIONS
REFERENCES (REVIEW)
GDB -DEMO

Problem Solving with Computers-II



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

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

https://ucsb-cs24-sp17.github.io/

Read the syllabus. Know what's required. Know how to get help.

CLICKERS OUT – FREQUENCY AB

Where are you with PA1?

A. Haven’t started

B. Less than 50% done

C. Almost done, have trouble testing my code

D. Done, passed almost all the test cases on submit

Announcements

- Extra open lab hours every week: Mondays 2pm to 4:00pm, 6:30pm to 8:00pm
- Reach out to your mentors!

Passing point objects as parameters (Review)

```
double distance(point p1, point p2);
```

//Precondition: p1 and p2 are point objects that have been initialized

//Post condition: returns the Euclidean distance between the two points

Would you implement the above function as a member function or a non-member function? Write your reason and discuss with your peer group.

A. Member function

B. Non-member function

C. Neither

Passing point objects as parameters (Review)

```
double distance(point p1, point p2);
```

//Precondition: p1 and p2 are point objects that have been initialized
//Post condition: returns the Euclidean distance between the two points

Which of the following is invoked when passing parameters to the distance function is (on line 2):

```
point s1(1,1), s2; //line 1
cout<<distance(s1, s2); //line 2
```

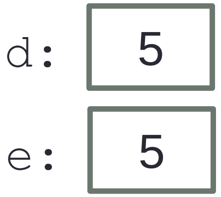
A. Default constructor
B. Default assignment operator
C. Default copy constructor

References in C++

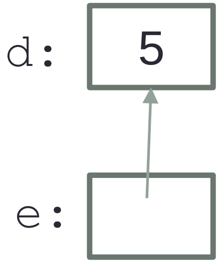
```
int main() {
  int d = 5;
  int &e = d;
}
```

Which diagram below represents the result of the above code?

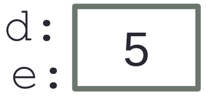
A.



B.



C.



D. This code causes an error

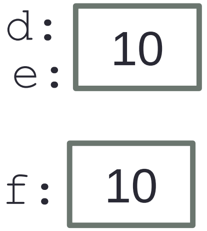
References in C++

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

}
```

How does the diagram change with this code?

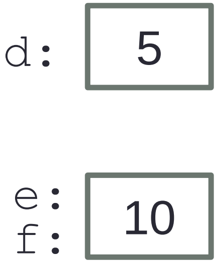
A.



d:
e: 10

f: 10

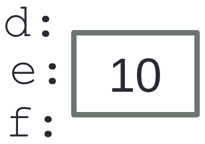
B.



d: 5

e:
f: 10

C.



d:
e:
f: 10

D. Other or error

Passing references as parameters

```
double distance(point &p1, point &p2);
```

//Precondition: p1 and p2 are point objects that have been initialized

//Post condition: returns the Euclidean distance between the two points

```
point s1(1,1), s2;
cout<<distance(s1, s2);
```

What is the benefit of passing references as parameters?

What are the potential dangers?

Operator overloading

In the previous class we overloaded the equality operator

==

bool operator ==(point p1, point p2); //function declaration

So we could use it in the following way:

```
double distance(const point & p1, const point &p2){
     if(p1 == p2)
       return 0;

}
```

Printing point objects to output stream

- By overloading the << and >> operators we could do the following :

```
point p(10, 10);
cout<<p;
```

And this….

```
point p;
cin>>p; //sets the x and y member variables of p based on user input
```

Demo

- New distance function
- Operator overloading and friend function wrap up
- Separate compilation with makefiles
- Debugging with gdb

Next time

- Container classes