DEFAULT PARAMETERS, OPERATOR OVERLOADING FRIEND FUNCTIONS

Problem Solving with Computers-II

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

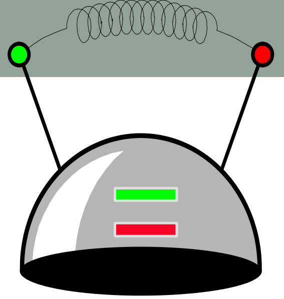
```
#include <iostream>
using namespace std;

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

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

CLICKERS OUT – FREQUENCY AB

Review: Constructor



Which constructor is called when the following statement is executed?

```
thinking_cap student;
```

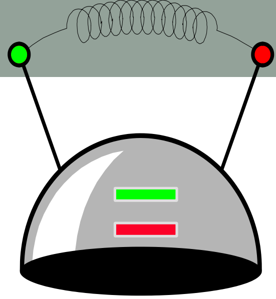
```
class thinking_cap
{
public:
    thinking_cap();                                    //A
    thinking_cap(char new_green[], char new_red[]); //B
    void slots(char new_green[ ], char new_red[ ]);
    void push_green( ) const;
    void push_red( ) const;
private:
    char green_string[50];           //C: Default copy constructor
    char red_string[50];             //D: Default assignment operator
};                                   //E: None of the above
```

Default values

```
int sum(int a=10, int b=20){
        return a+b;
}

int main(){
        int x= 40, y=50;
        cout<<sum(x,y)<<endl;
        cout<<sum(x)<<endl;
        cout<<sum()<<endl;
}
```

Specify default constructor using default arguments



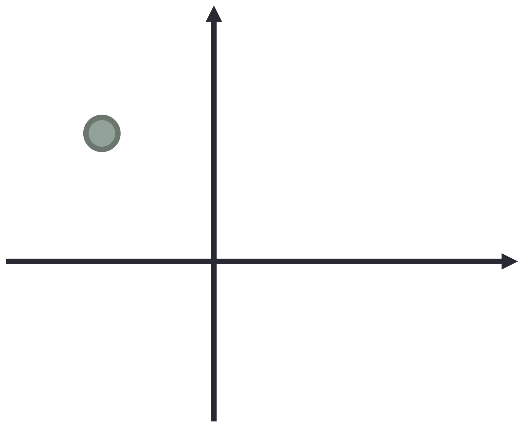
Which constructor is called when the following statement is executed?

```
thinking_cap student;
```

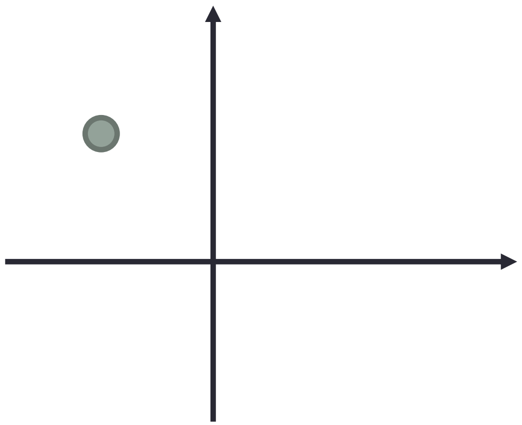
```
class thinking_cap
{
public:
    thinking_cap(char new_green[]=“Hello”, char new_red[]=“there”); //A
    void slots(char new_green[ ], char new_red[ ]);
    void push_green( ) const;
    void push_red( ) const;
private:
    char green_string[50];
    char red_string[50];
};
```

//B: Default copy constructor
//C: Default assignment operator
//D: None of the above

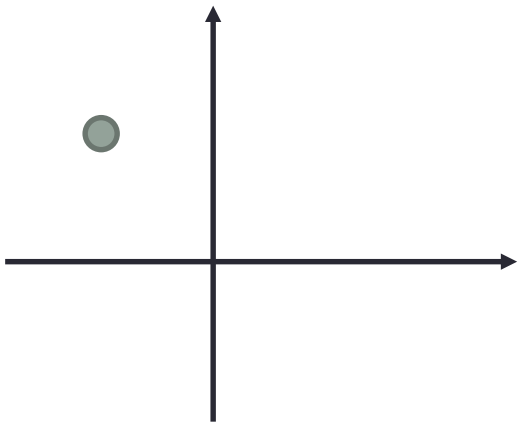
The point class (Chapter 2, section 2.4)



point: (x,y)



shift(delx, dely)



rotate90()

Let’s look at the implementation of the point class

Passing point objects 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

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

```
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 the distance function is called on s1 and s2 (line 2):

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

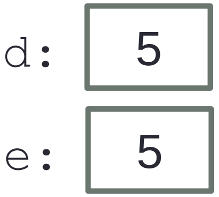
A. Default constructor
B. Default assignment operator
C. Default copy constructor

References in C++

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

Which diagram below represents the result of the above code?

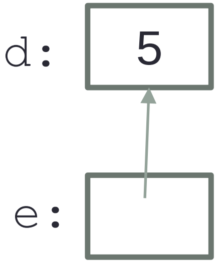
A.



d: 5

e: 5

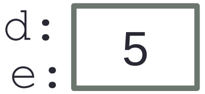
B.



d: 5

e:

C.



d:

e: 5

D. This code causes an error

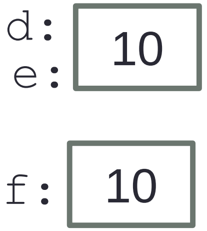
References in C++

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

}
```

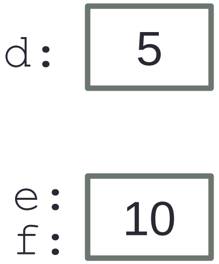
How does the diagram change with this code?

A.


d:
e:
10

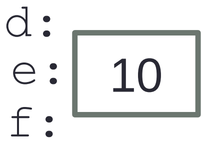
f:
10

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 potential dangers?

Operator overloading

We would like to be able to compare two objects of the class using the following operators

==

!=

and possibly others

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

}
```

Printing point objects to output stream

- Wouldn’t it be convenient if we could do this:

```
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
```

Summary

- Classes have member variables and member functions (method). An object is a variable where the data type is a class.
- You should know how to declare a new class type, how to implement its member functions, how to use the class type.
- Frequently, the member functions of an class type place information in the member variables, or use information that's already in the member variables.
- New functionality may be added using non-member functions, friend functions, and operator overloading

Next time

- Wrap up chapter 2, gdb