OPERATOR OVERLOADING

Problem Solving with Computers-II

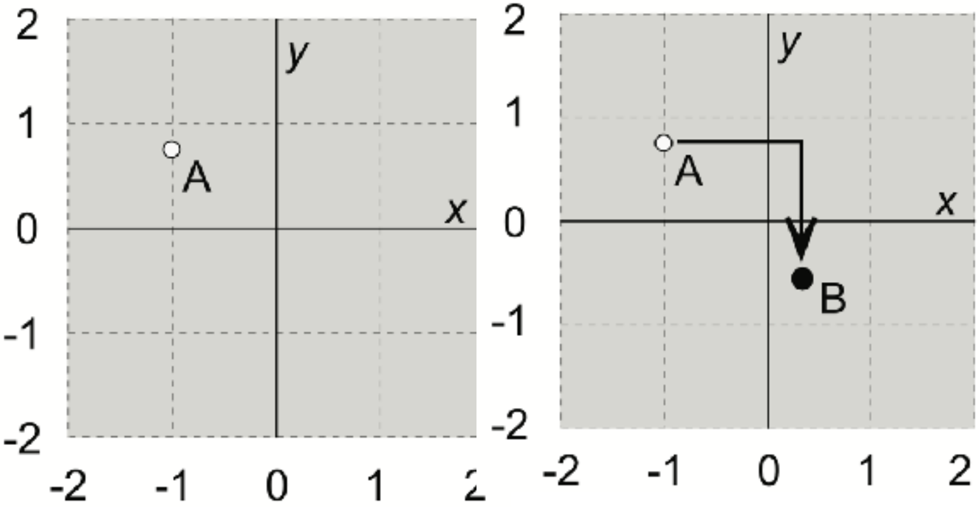
```
#include <iostream>
using namespace std;

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

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

CLICKERS OUT

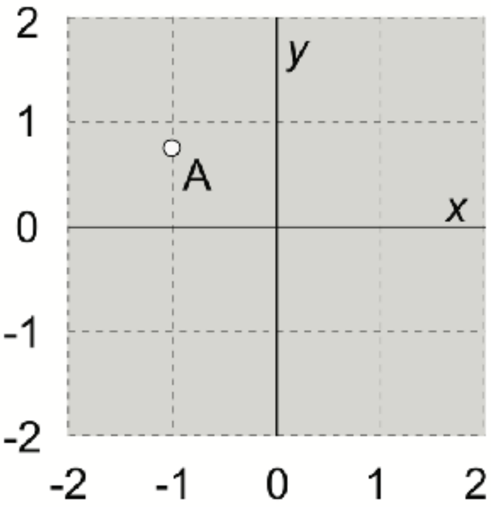
The point class (Chapter 2, section 2.4)

(a) The white dot labeled A is a point with coordinates $x = -1.0$ and $y = 0.8$.



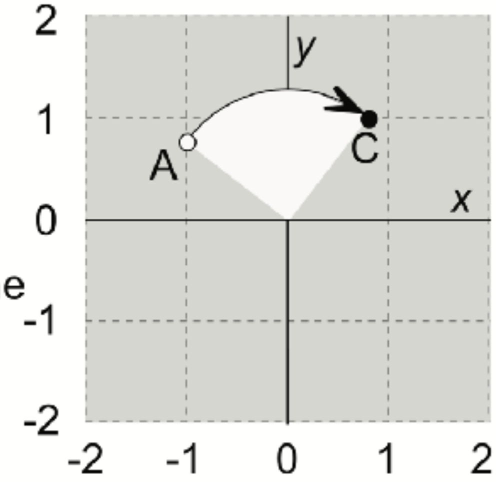
(b) The black dot labeled B was obtained by shifting point A by 1.3 units along the x axis and by -1.4 units along the y axis. The coordinates of point B are $x = 0.3$ and $y = -0.6$.

The point class (Chapter 2, section 2.4)



(a) The white dot labeled A is a point with coordinates $x = -1.0$ and $y = 0.8$.

(c) The black dot labeled C was obtained by rotating point A 90° in a clockwise direction around the origin. The coordinates of point C are $x = 0.8$ and $y = 1.0$.



Overloading Binary Comparison Operators

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;

}
```

Overloading Binary Arithmetic Operators

We would like to be able to add two points as follows

```
point p1, p2;
point p3 = p1 +p2
```

Overloading input/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

Copy assignment

```
Player q;         //default constructor is invoked
Player p1(“Jill”); // Parametrized constructor
Player p2;
p2 = p1; // Copy assignment method is invoked
```

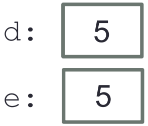
- Default behaviour: Member variables of p1 are copied to the members variables are p2

References in C++

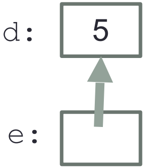
```
int main() {
  int d = 5;
  int &e = d;
}
```

Which diagram below represents the result of the above code?

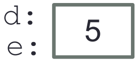
A.



B.



[✓ C.]



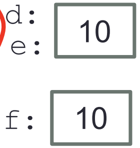
D. This code causes an error

References in C++

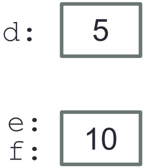
```
int main() {
   int d = 5;
   int &e = d;
   int f = 10;
   e = f;

}
```

How does the diagram change with this code?

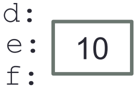
[✓ A.]



B.



C.



D. Other or error

Passing parameters as references

```
int main() {
   int d = 5;
   foo(d);
   cout<<d;
}
```

```
void foo(int& e) {
   e = 10;
}
```

What is the output of this code?

A. 5

[✓ B. 10]

C. Error

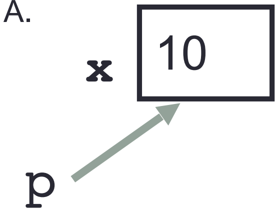
D. None of the above

Tracing code involving pointers

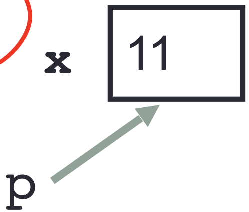
```
int *p, x = 10;
p = &x;
*p = *p + 1;
```

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

A.



[✓ B.]



C. Neither, the code is incorrect

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

- Linked-lists (Chapter 5)