REVIEW: REFERENCES, BIG FOUR OPERATOR OVERLOADING

Problem Solving with Computers-II

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

CLICKERS OUT

References in C++

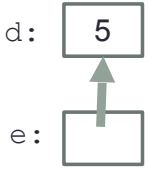
```
int main() {
  int d = 5;
  int &e = d;
}
```

e is an alias for 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;

}
```

changing e is the same as changing d because..... e is an alias for d

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 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

Typical use of references when passing parameters to function.

We use references as parameters either for efficiency (avoids copying the data) OR to modify data that is out of the scope of the function

Copy constructor (Review)

- In which of the following cases is the copy constructor called?

```
A. Player p1; Player p2("Jill");
B. Player p1("Jill"); Player p2(p1);
C. Player *p1 = new Player("Jill"); Player p2 = *p1;
D. B&C
E. A, B & C
```

Copy constructor (Review)

- The copy constructor creates and initializes a new object to be the copy of another object of the class
- C++ provides a default copy constructor if one is not defined in the definition of the class
- The copy constructor is called in all the following cases, assuming p1 is an existing object of Player:

```
Player p2(p1);
Player p2 = p1;
Player *p2 = new Player(p1);
```

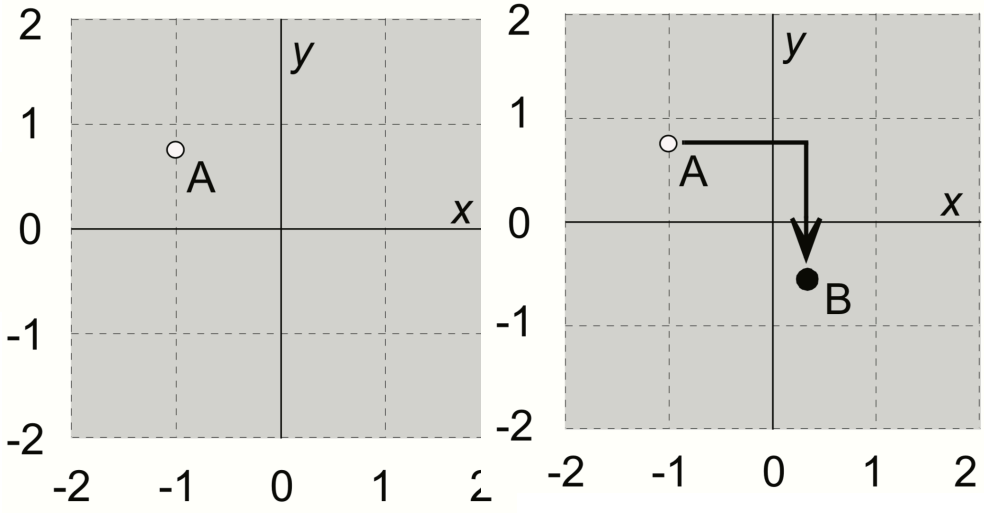
Copy assignment

- Default behavior: Copies the member variables of one object into another

```
Player p1("Jill"); // Parametrized constructor
Player p2;
p2 = p1; // Copy assignment function is called
```

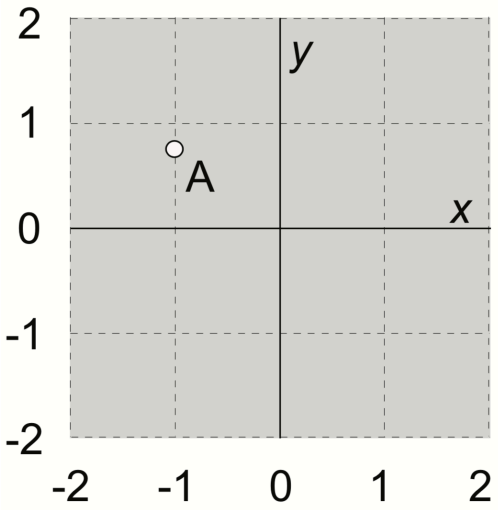
The point class (Chapter 2, section 2.4)

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



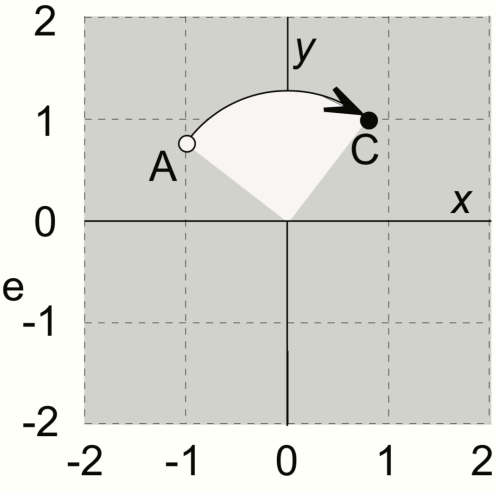
(b) The black dot labeled B was obtained by shifting point A by 1.3 units along the x axis and by -1.4 units along the y axis. The coordinates of point B are $x = 0.3$ and $y = -0.6$.

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;

}
```

Please refer to code written in lecture

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
```

Next time

- Linked-lists (Chapter 5)