Discussion 9!

Project 3, Operator Overloading

Project 3!!

Read Image In

```
P3
4 4          width and height
255          max value
0 0 255    0 0 255    0 0 255    0 0 255
0 0 255    255 0 0    255 0 0    0 0 255
0 0 255    255 0 0    255 0 0    0 0 255
0 0 255    0 0 255    0 0 255    0 0 255
```

.ppm image is read in

“P3” is the ‘magic number’

Next row is the number of rows and cols

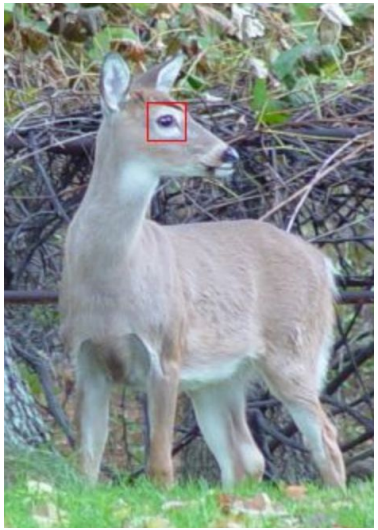
Next number is the max color value (255)

Then all the pixels are listed out (rows * cols * 3) in RGB order

Store the images in a dynamically allocated 2D array

Error check, error check, error check!!!

Modifying Images- Rectangle



Filled rectangles or outlines placed over image

specifying the rectangle via:

1) the upper-left and lower-right locations directly

2) specifying the upper-left corner and a width and height

3) specifying the center of a rectangle and a width extent and height extent from the center (i.e. half-width and half-height)

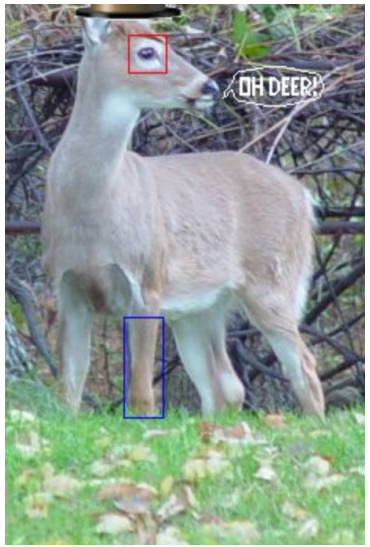
Modifying Images- Pattern from file

Reads in rows in columns

Then reads in rectangle of 1s and 0s

1s represent where to color over

0s represent where to keep the color the same



rows cols

6 8

1 1 1 1 1 1

1 1 1 1 1 1

0 0 1 1 0 0

0 0 1 1 0 0

0 0 1 1 0 0

0 0 1 1 0 0

0 0 1 1 0 0

0 0 1 1 0 0

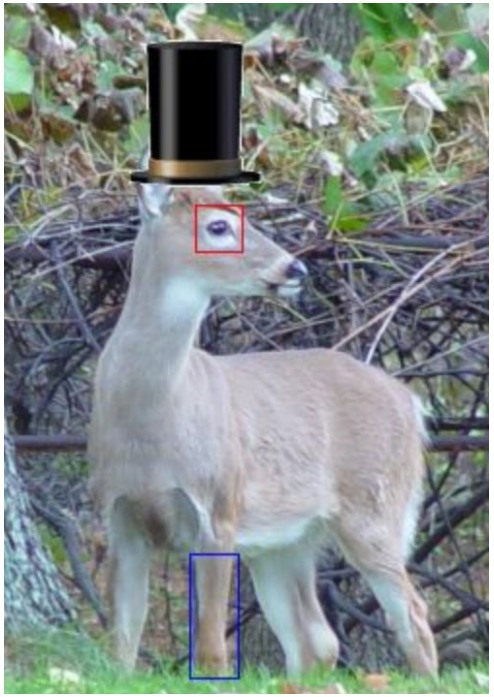
Modifying Images- Image Insertion

Reads a (smaller) .ppm file in

Inserts the file where to user specifies

Uses transparency color

The color that you ignore



Printing Image

Allows you to test your program, recommend doing this Page3

Reads the output filename in from the user

Prints the current image to an output file

Put in valid ppm format

Viewing Images

Convert to JPEG- (command typed on Linux)

```
%  cjpeg inFile.ppm > outFile.jpg
```

Or

```
% convert inFile.ppm outFile.jpg
```

Or

Download a program that can open and view .ppm files (IrfanView is one example)

Tips

- Be careful of magic numbers
- Read the implementation and design part of spec!
- Where to start?
- Please start if you haven’t!
- Read Piazza for details on error checking!

Operator Overloading

Operator Overloading

Allows us to use operators on data types that wouldn't usually be able to use operators

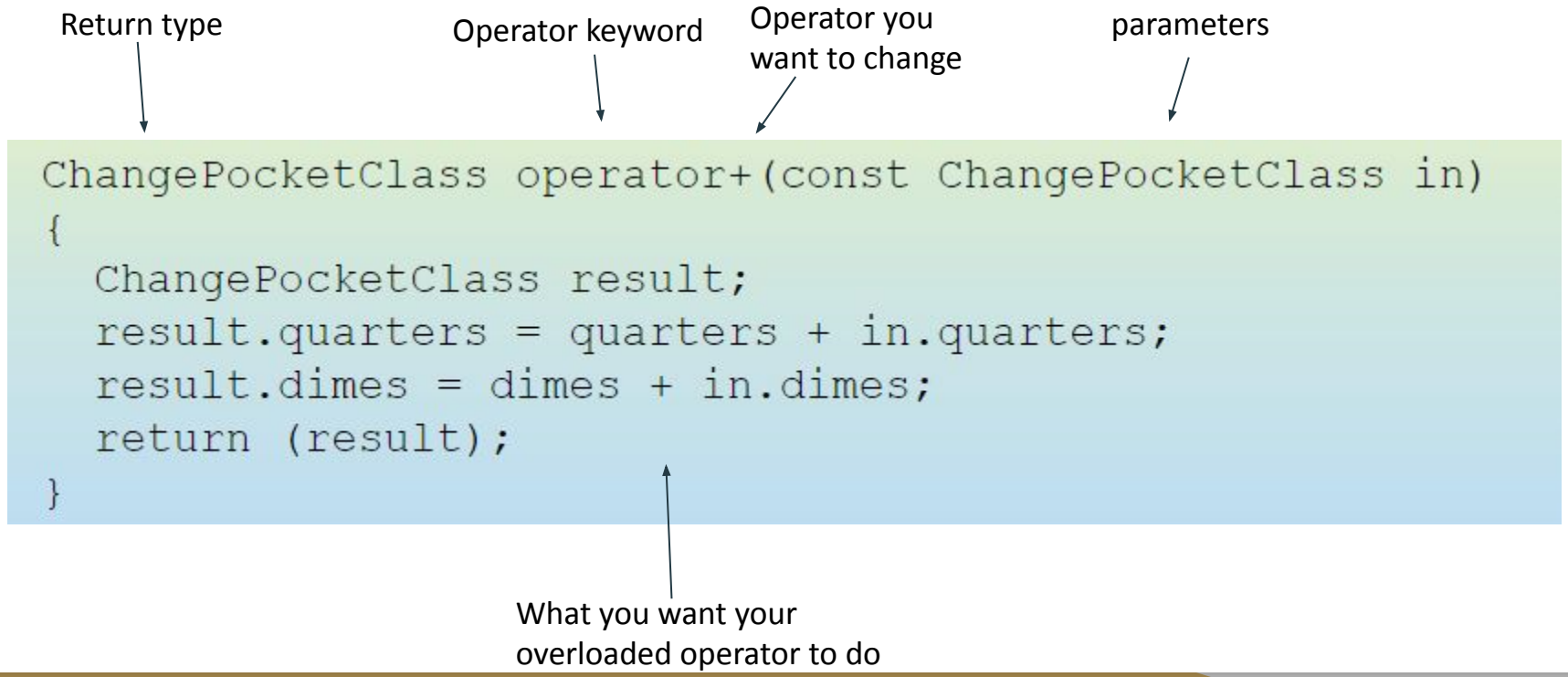
Most common operators overloaded: +, -, =, <, >, ==, <<

The following list of operators can be overloaded:

+	-	*	/	%	^	&	|	~
!	=	<	>	+=	-=	*=	?=	%=
^=	&=	|=	<<	>>	>>=	<<=	==	!=
<=	>=	&&	||	++	--	->*	,	->
[]	()	new	delete	new[]	delete[]			

if(redColor>blueColor)

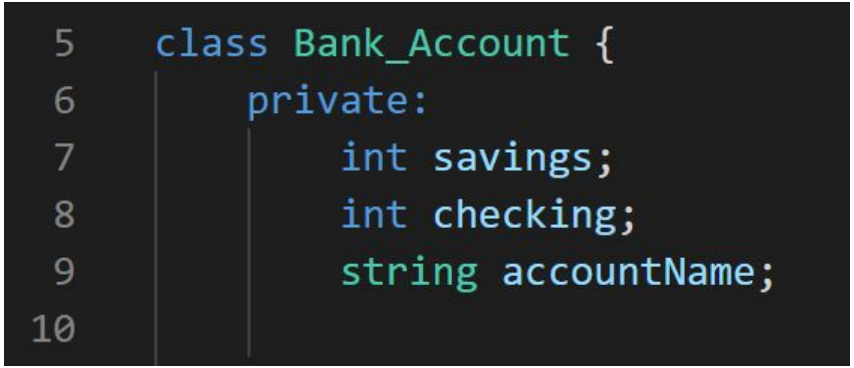
Components of Overload Function



Operator Overloading Example

Overload functions for this Bank_Account class

What operations might be helpful?



.cpp file: https://drive.google.com/file/d/15CNvq1AFn5qwuer8P5ei0srR-K-s6HW8/view?usp=sharing

git clone https://github.com/emolson16/overloadOperator.git

Operator Overloading Example

Overload functions for this Bank_Account class

What operations might be helpful?

+, ==, +=, <, -=

```
5   class Bank_Account {
6       private:
7           int savings;
8           int checking;
9           string accountName;
10
```

.cpp file: https://drive.google.com/file/d/15CNvq1AFn5qwuer8P5ei0srR-K-s6HW8/view?usp=sharing

git clone https://github.com/emolson16/overloadOperator.git

Bank_Account Overloading Sample Solution

```
18          Bank_Account(){};
19
20          bool operator==(const Bank_Account & other) {
21              return ((other.savings + other.checking) == (savings + checking));
22          }
23
24          bool operator<(const Bank_Account & other) {
25              return ((savings + checking) < (other.savings + other.checking));
26          }
27
28          void operator+=(int money){
29              savings += money;
30          }
31
32          void operator-=(int money){
33              savings -= money;
34          }
35
36          Bank_Account operator+(const Bank_Account & other) {
37              Bank_Account result;
38              result.savings = other.savings + savings;
39              result.checking = other.checking + checking;
40              return result;
41          }
42
```