# Exercise 1:

#include <iostream>

using namespace std;

// Create new class for Banana Crates

class Banana\_Crate

{

// Public modifier to be accessed outside of class

public:

// Properties

int MAX\_NUM\_BANANAS = 5; // Default banana max if none is declared

int numOfBananasTotal = 0; // Keeps current total of bananas for crate

int maxNumBananas = 0; // If max count for crate is specified then assign to this value

int bananaCountInBox = 0; // Tracks the total in current crate

// Default constructor

Banana\_Crate()

{

maxNumBananas = MAX\_NUM\_BANANAS;

}

// Override defult constructor

Banana\_Crate(int maxBananas)

{

maxNumBananas = maxBananas;

}

// Methods

void addBanana()

{

// If the current banana count is more or equal to the max the crate can hold

// create a new crate with the value of one and add to the total of bananas

if (bananaCountInBox >= maxNumBananas)

{

bananaCountInBox = 1;

numOfBananasTotal++;

}

// Otherwise add to both totals

else

{

bananaCountInBox++;

numOfBananasTotal++;

}

}

// Display the current banana count in box and total

void getCurrentBananaCount()

{

cout << "The number of bananas in the box is: " << bananaCountInBox << endl;

cout << "The current number of bananas is: " << numOfBananasTotal << endl;

}

// Empty the crate but keep current count

void emptyCrate()

{

bananaCountInBox = 0;

}

};

int main()

{

// Create two new instances of the crate class and override one with a new default max size

Banana\_Crate crate1;

Banana\_Crate crate2(3);

// Add to crate 2

crate2.addBanana();

crate2.addBanana();

crate2.addBanana();

crate2.addBanana();

crate2.addBanana();

// Add to crate 1

crate1.addBanana();

crate1.addBanana();

cout << "For create 1 " << endl;

crate1.getCurrentBananaCount();

cout << "For create 2 " << endl;

crate2.getCurrentBananaCount();

}

A screenshot of a computer

Description automatically generated with medium confidence