

```

12     void PrintMaterial() const {
13         cout << name << " - " << density << endl;
14     }
15     string GetName() const { return name; }
16     int GetDensity() const { return density; }
17
18     private:
19         string name;
20         int density;
21     };
22
23 int main() {
24     vector<Material> requestedMaterials;
25     Material currMaterial;
26     string currName;
27     int currDensity;
28     unsigned int i;
29
30     cin >> currName;
31     cin >> currDensity;
32     while ((currName != "quit") && (currDensity > 0)) {
33
34     /* Your code goes here */
35

```

©zyBooks 10/01/23 21:16 191741  
 John DeGood  
 TCNJCSC215DeGoodSpring2022

1

2

3

[Check](#)

[Next](#)

[View solution](#) ▾ (*Instructors only*)

## 7.8 Separate files for classes

### Two files per class

Programmers typically put all code for a class into two files, separate from other code.

- **ClassName.h** contains the class definition, including data members and member function declarations.
- **ClassName.cpp** contains member function definitions.

©zyBooks 10/01/23 21:16 191741  
 TCNJCSC215DeGoodSpring2022

A file that uses the class, such as a main file or ClassName.cpp, must include ClassName.h. The .h file's contents are sufficient to allow compilation, as long as the corresponding .cpp file is eventually compiled into the program too.

The figure below shows how all the .cpp files might be listed when compiled into one program. Note that the .h file is not listed in the compilation command, due to being included by the appropriate .cpp files.

Figure 7.8.1: Using two separate files for a class.

StoreItem.h

```
#ifndef STOREITEM_H
#define STOREITEM_H

class StoreItem {
public:
    void SetWeightOunces(int
ounces);
    void Print() const;
private:
    int weightOunces;
};

#endif
```

StoreItem.cpp

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

#include "StoreItem.h"

void StoreItem::SetWeightOunces(int ounces) {
    weightOunces = ounces;
}
John DeGood
TCNJCSC215DeGoodSpring2022

void StoreItem::Print() const {
    cout << "Weight (ounces): " << weightOunces
    << endl;
}
```

main.cpp

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

#include "StoreItem.h"

int main() {
    StoreItem item1;

    item1.SetWeightOunces(16);
    item1.Print();

    return 0;
}
```

Compilation example

```
% g++ -Wall StoreItem.cpp main.cpp
% a.out
Weight (ounces): 16
```

## Good practice for .cpp and .h files

Sometimes multiple small related classes are grouped into a single file to avoid a proliferation of files. But for typical classes, good practice is to create a unique .cpp and .h file for each class.

©zyBooks 10/01/23 21:16 191741

John DeGood

TCNJCSC215DeGoodSpring2022

### PARTICIPATION ACTIVITY

7.8.1: Separate files.



- 1) Commonly a class definition and associated function definitions are placed in a .h file.

True  
 False

- 2) The .cpp file for a class should #include the associated .h file.

True  
 False

- 3) A drawback of the separate file approach is longer compilation times.

- True
- False

## Ex: Restaurant review classes

©zyBooks 10/01/23 21:16 191741

John DeGood

TCNJCSC215DeGoodSpring2022

The restaurant review program, introduced in an earlier section, declared the Review, Reviews, and Restaurant classes in main.cpp. Each of the 3 classes should instead be implemented in .h/.cpp files, thus making for cleaner code in main.cpp.

Figure 7.8.2: .h and .cpp files for Review, Reviews, and Restaurant classes.

Review.h

```
#ifndef REVIEW_H
#define REVIEW_H

#include <string>

class Review {
public:
    void SetRatingAndComment(
        int revRating,
        std::string
revComment);
    int GetRating() const;
    std::string GetComment() const;

private:
    int rating = -1;
    std::string comment =
"NoComment";
};

#endif
```

Review.cpp

```
#include "Review.h"
using namespace std;

void Review::SetRatingAndComment(int
revRating, string revComment) {
    rating = revRating;
    comment = revComment;
}

int Review::GetRating() const {
    return rating;
}

string Review::GetComment() const {
    return comment;
}
```

©zyBooks 10/01/23 21:16 191741

John DeGood

TCNJCSC215DeGoodSpring2022

## Reviews.h

```
#ifndef REVIEWS_H
#define REVIEWS_H

#include <vector>
#include "Review.h"

class Reviews {
public:
    void InputReviews();
    void PrintCommentsForRating(int currRating) const;
    int GetAverageRating() const;

private:
    std::vector<Review> reviewList;
};

#endif
```

## Reviews.cpp

```
#include <iostream>
#include "Reviews.h"
using namespace std;

// Get rating comment pairs, add each to list.
-1 rating ends.
void Reviews::InputReviews() {
    Review currReview; ©zyBooks 10/01/23 21:16 191741
    int currRating; John DeGood
    string currComment; TCNJCSC215DeGoodSpring2022

    cin >> currRating;
    while (currRating >= 0) {
        getline(cin, currComment); // Gets rest
        of Line

        currReview.SetRatingAndComment(currRating,
        currComment);
        reviewList.push_back(currReview);
        cin >> currRating;
    }
}

// Print all comments for reviews having the
given rating
void Reviews::PrintCommentsForRating(int currRating) const {
    Review currReview;
    unsigned int i;

    for (i = 0; i < reviewList.size(); ++i) {
        currReview = reviewList.at(i);
        if (currRating ==
currReview.GetRating()) {
            cout << currReview.GetComment() <<
endl;
        }
    }
}

int Reviews::GetAverageRating() const {
    int ratingsSum;
    unsigned int i;

    ratingsSum = 0;
    for (i = 0; i < reviewList.size(); ++i) { ©zyBooks 10/01/23 21:16 191741
        ratingsSum += reviewList.at(i).GetRating(); John DeGood
    }
    return (ratingsSum / reviewList.size());
}
```

## Restaurant.h

```
#ifndef RESTAURANT_H
#define RESTAURANT_H

#include <string>
#include "Reviews.h"

class Restaurant {
public:
    void SetName(std::string restaurantName);
    void ReadAllReviews();
    void PrintCommentsByRating() const;

private:
    std::string name;
    Reviews reviews;
};

#endif
```

## Restaurant.cpp

```
#include <iostream>
#include "Restaurant.h"
using namespace std;

void Restaurant::SetName(string restaurantName) {
    name = restaurantName;
}

void Restaurant::ReadAllReviews() {
    cout << "Type ratings + comments. To end:
-1" << endl;
    reviews.InputReviews();
}

void Restaurant::PrintCommentsByRating() const {
    int i;

    cout << "Comments for each rating level: "
<< endl;
    for (i = 1; i <= 5; ++i) {
        cout << i << ":" << endl;
        reviews.PrintCommentsForRating(i);
    }
}
```

### PARTICIPATION ACTIVITY

7.8.2: Restaurant reviews program's main.cpp.



### Animation content:

undefined

### Animation captions:

1. The Review, Reviews, and Restaurant classes are included in main.cpp by including Restaurant.h.
2. main()'s code is reasonably short, since reusable code resides in external files.

### PARTICIPATION ACTIVITY

7.8.3: Restaurant review program .h and .cpp files.



If unable to drag and drop, refresh the page.

[Reviews.h](#)

[Review.cpp](#)

[Review.h](#)

[Restaurant.cpp](#)

[Reviews.cpp](#)

[Restaurant.h](#)

#includes the "Restaurant.h" header file.

Uses cin and getline() statements to get ratings and comments from the user.

Makes the Restaurant, Reviews, and Review classes available when being #included by another code file.

Does not #include any of the 3 header files.

#includes the <vector> header file.  
©zyBooks 10/01/23 21:16 191741  
John DeGood  
TCNJCSC215DeGoodSpring2022

Implements class member functions, none of which use cin or cout.

Reset

CHALLENGE ACTIVITY

7.8.1: Enter the output of separate files.



383092.383482.qx3zqy7

Start

Type the program's output

main.cpp Product.h Product.cpp

```
#include <iostream>
#include <vector>
#include "Product.h"
using namespace std;

int main() {
    vector<Product> productList;
    Product currProduct;
    int currPrice;
    string currName;
    unsigned int i;
    Product resultProduct;

    cin >> currPrice;
    while (currPrice > 0) {
        cin >> currName;
        currProduct.SetPriceAndName(currPrice, currName);
        productList.push_back(currProduct);
        cin >> currPrice;
    }

    resultProduct = productList.at(0);
    for (i = 0; i < productList.size(); ++i) {
        if (productList.at(i).GetPrice() < resultProduct.GetPrice()) {
            resultProduct = productList.at(i);
        }
    }

    cout << resultProduct.GetName() << ":" << resultProduct.GetPrice() << endl;
}

return 0;
}
```

Input

5 Berries  
6 Foil  
12 Shirt  
-1

©zyBooks 10/01/23 21:16 1  
John DeGood  
TCNJCSC215DeGoodSpring2022

Output

Berries: 5

1

2

3

Check

Next

View solution ▾ (Instructors only)