Yisroel Appleton

Madeleine Kobus

Alexander McKinney

Garrett Penfield

Garrett and Alex: coded the classes at the end of this pseudocode.

Tina and Madeleine: worked on the outline of events/pseudocode and created the diner menu

9/26: This week work on creating the vectors to hold the food and drink items of the menu. Combine the classes we already have, and work them into an online code share/work space. Code the output dialogue for the waitress.

10/3: Fine tune the inputs from the user. If statements for Table or Booth. Do you want to see a menu? Y or N. Create a printed out menu for the user to view. These will be split up by group members and worked into the online share/work space.

10/10: Test project using multiple situations (i.e testing limits on inputs, test . Debug). Fix any errors or make any changes that do not work with the flow of the program.

10/17: Turn in project

Diner Simulator

out “How many are in your party?”

if (number > 1 & <= 10) ok

out “Table or Booth”

Enter T for Table or B for Booth

out “Welcome to your (Table or Booth), I am Rita and I will be your server today.”

out “What are all of your names?

in names

out “Can I get you started off with some drinks?”

Enter Y or N

if N (ask if they want to order or see menu?)

out “We have water (1), coffee (2), hot tea (3), (4) iced tea, (5) lemonade, (6) hot coco, and (7) soda.”

out “would you like to see our list of sodas?”

(Y or N)

if Y(8 Pepsi, 9 Mr. Dew, 10 Sprite…hit any key to close menu

in >> Drink number

out “Thanks! I’ll bring those right out. In the mean time, why don’t you have a look at the menu?”

// Display menu press any key to begin ordering your food.

out “Person1 (by name), what can I get for you today…..

in Category = B for Burger

in Subcategory = # of Burger

in Extras = add cheese, no pickle, etc…

out “Person2, what can I get for you today….

in Category = R for Breakfast

in Subcategory = # option

in Substitutes = sausage for bacon, etc…

etc…

food is brought out…

out “Can I get you anything else?

K for Ketchup

N for no

etc…

get check….

calculates itemized receipt.

Person1 = cheeseburger with fries, add pickles, chocolate shake, and water = total + tax.

Person2 = French Toast meal with coffee and cherry pie = total + tax

out “Do you want to leave a tip?”

Y or N

if Y( 10% is $x, 15% is $y, 20% $z)

print total bill with tip , tax, total.

DRAFT CODE:

Class Restaurant

{

private:

    string restaurantName;

    string serverName;

    int capacity;

public:

void setRestaurantName(string \_restaurantName);

void setServerName(string \_serverName);

void setCapacity(int \_capacity);

string getRestaurantName() const;

string getServerName() const;

int getCapacity() const;

};

void setRestaurantName(string \_restaurantName){

restaurantName = \_restaurantName;

};

void setServerName(string \_serverName){

serverName = \_serverName;

};

void setCapacity(int \_capacity){

capacity = \_capacity;

};

string getRestaurantName(){

return restaurantName

};

string getServerName(){

return serverName;

};

int getCapacity(){

return capacity;

};

Class orderItem

{

private:

    string customerName, serverName;

    boolean paid;

    double total, tax, tip;

    int tableNumber, serverRating, numOfCustomers;

public:

void setCustomerName(string \_name);

void setServerName(string \_name);

void setPaid(boolean \_paid);

void setTotal(double \_total);

void setTax(double \_tax);

void setTip(double \_tip);

void setTableNumber(int \_tableNumber);

void setServerRating(int \_serverRating);

void setNumOfCustomers(int \_numOfCustomers);

string getCustomerName() const;

string  getServerName() const;

boolean getPaid() const;

double getTotal() const;

double getTax() const;

double getTip() const;

int getTableNumber() const;

int getServerRating() const;

int getNumOfCustomers() const;

};

 void setCustomerName(string \_name){

  customerName = \_name;

 };

void setServerName(string \_name){

serverName = \_name;

};

void setPaid(){

if (paid = false){

paid = true;

}

else{

paid = false;

}

};

void setTotal(double \_total){

total = \_total;

};

void setTax(double \_tax){

tax = \_tax;

};

void setTip(double \_tip){

tip = \_tip;

};

void setTableNumber(int \_tableNumber){

tableNumber = \_tableNumber;

};

void setServerRating(int \_serverRating){

serverRating = \_serverRating;

};

void SetNumOfCustomers(int \_numOfCustomers){

numOfCustomers = \_numOfCustomers;

};

string getCustomerName(){

return customerName;

};

string  getServerName(){

return serverName;

};

boolean getPaid(){

return paid;

};

double getTotal(){

return total;

};

double getTax(){

return tax;

};

double getTip(){

return tip;

};

int getTableNumber(){

return tableNumber;

};

int getServerRating(){

return serverRating;

};

int getNumOfCustomers(){

return numOfCustomers;

};

Class foodItem

{

private:

    string name, type, notes;

    double price;

public:

void setName(string \_name);

void setType(string \_type);

void setnotes(string \_note);

string getName() const;

string getType() const;

string getnotes() const;

};

void setName(string \_name){

name = \_name;

};

void setType(string \_type){

type = \_type;

};

void setnotes(string \_note){

note = \_note;

};

string getName(){

return name;

};

string getType(){

return type;

};

string getnotes(){

return notes;

};

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Breakfast** | | |  |  | **Pie (by the slice)** |  |
| **1** | **2 Eggs and Toast** | **$2.99** |  |  | **Apple Pie** | **$1.99** |
| **2** | **2 Eggs Hash browns, Toast, Bacon** | **$4.99** |  |  | **Cherry Pie** | **$1.99** |
| **3** | **Biscuits & Gravy with 2 eggs** | **$6.99** |  |  | **Pumpkin Pie** | **$1.99** |
| **4** | **Eggs Benedict with Fruit** | **$5.99** |  |  | **Pecan Pie** | **$1.99** |
| **5** | **Western Omelet** | **$8.99** |  |  | **Chocolate Pie** | **$1.99** |
| **6** | **Country Omelet** | **$7.99** |  |  | **Blueberry Pie** | **$1.99** |
| **7** | **USDA Choice Serloin & 2 Eggs** | **$9.99** |  |  | **Al la mode?** | **$0.50** |
| **8** | **Corned Beef Hash & 2 Eggs** | **$6.99** |  |  |  |  |
| **9** | **Pancakes, 2 Eggs, and Bacon** | **$7.99** |  |  |  |  |
| **10** | **French Toast, 2 Eggs, and Bacon** | **$7.99** |  |  | **Drinks** |  |
|  |  |  |  |  | **Coffee** | **$1.00** |
| **Burgers** | | |  |  | **Iced Tea** | **$1.50** |
| **11** | **Classic Hamberger with Fries** | **$5.99** |  |  | **Hot Tea** | **$1.50** |
| **12** | **Cheeseburger with Fries** | **$6.99** |  |  | **Hot Cocoa** | **$1.50** |
| **13** | **Double Cheeseburger with Fries** | **$8.99** |  |  | **Soda** | **$2.00** |
| **14** | **Grilled Chicken Burger with Fries** | **$9.99** |  |  | **Water** | **$0.00** |
| **15** | **Pork Tenerloin with Fries** | **$7.99** |  |  |  |  |
|  |  |  |  |  |  |  |
| **Comfort Food** | | |  |  | **Shakes** |  |
| **16** | **Beef & Noodles with Mash Potatoes & Gravy** | **$10.99** |  | **20** | **Strawberry** | **$2.50** |
| **17** | **Hot Beef Sandwhich with Mashed Potatoes & Gravy** | **$10.99** |  | **21** | **Chocolate** | **$2.50** |
| **18** | **Meatloaf with Mashed Potatoes & Gravy** | **$9.99** |  | **22** | **Rootbeer** | **$2.50** |
| **19** | **Fried Chicken with Mashed Potatoes & Gravy** | **$10.99** |  | **23** | **Vanilla Bean** | **$2.50** |