Object Cheat Sheet

6.2 K

Name: Solution

Example	What?	Method Signature Notes	Inside the method Notes
public class Item { private double price; private String name;	Class opening line - instance variables		• Class name is singular and has a capital letter at the front. • private in front of instance variables.

public Item(){
 price = 13.45;
 name = "t-shirt";
 }
	default constructor (called with new)	Return type: none (technically itself)
		 Name: same as class
		 Parameters: none
			• Set each instance variable = to a reasonable value

public Item(double p, String n){
 price = p;
 name = n;
 }
	customized constructor (called with new)	Return type: none (technically itself)
		 Name: same as class
		 Parameters: one for each instance variable
			• Set each instance variable = to its corresponding parameter

public void setPrice(double p){
 price = p;
 }
 public void setName (String n){
 name = n;
 }
	mutators (sets)	Return type: void
		 Name: set + instance variable
		 Parameters: same type as instance variable
			• Set ~~each~~ instance variable = to parameter

public double getPrice(){
 return price;
 }
 public String getName(){
 return name;
 }
	accessors (gets)	Return type: same type as instance variable
		 Name: get + instance variable
		 Parameters: none
			• Return the instance variable

public String toString(){
 return "The "+name+" costs $"+price;
 }
	toString (an accessor)	Return type: String
		 Name: toString
		 Parameters: none
			• Return a sentence which links up all of the instance variables

public boolean equals(Item i){
 if(i.getName().equals(name)
 && i.getPrice()==price)
 return true;
 else
 return false;
 }
	equals (facilitator)	Return type: boolean
		 Name: equals
		 Parameters: same type as class (recursive parameter)
			• Check each parameter's accessor against each instance variable.
			 • If all match, return true. Otherwise return false.

public int compareTo(Item i){
 //on the basis of price
 if(i.getPrice()>price)
 return -1;
 else if (i.getPrice()==price)
 return 0;
 else
 return 1; }
 }
	compareTo (facilitator)	Return type: int
		 Name: compareTo
		 Parameters: same type as class (another recursive parameter)
			• Pay attention to what you are comparing. It changes based on the question.
			 • Check the parameter's accessor against the instance variable.
			 • If (param>instance) return -1
			 • Else if (param==instance) return 0
			 • Else return 1