Submit answers for all of the following questions.

EXERCISES:

Chapter 13:

10.

a) doIt is being called inside of itself and because of that it will run in an infinite loop.

b) To fix this it should be changed to super.doIT()

14. Final prevents any method from overwriting or changing the final one in any of the subclasses.

15. If it is used in a class it will prevent the class from having any subclasses.

17.

|  |  |  |
| --- | --- | --- |
| Structural member | Beam | Inheritance |
| Building | Floor | Composition |
| Company | Fixed Assets | Composition |
| Employee | Salesperson | Inheritance |
| Forest | Tree | Composition |
| Bird | Robin | Inheritance |
| Class | Method | Composition |
| Neurosis | Paranoia | Inheritance |

ADDITIONAL EXERCISES:

The following seven questions refer to the *Using the Eclipse Debugger* tutorial, which you should obtain from the course website.

1. Code

**public** **class** Mouse

{

**private** String variety; // standard, hairless, etc.

**private** **int** weight; // the mouse's weight in grams

**private** **double** growthRate; // % that mouse grows each day

**private** **int** days; // # of simulated days so far

**public** Mouse(String v, **int** w, **int** gR)

{

variety = v; weight = w; growthRate = gR;

} // end Mouse constructor

**public** **void** simulateGrowth(**int** d)

{

days += d;

**while** ((d > 0) && (weight > 0))

{

weight += growthRate/100 \* weight;

d--;

}

} // end simulateGrowth

**public** **void** printStatus()

{

System.***out***.println("After " + days + " days, " +

" weight = " + weight + ".");

} // end printStatus

**public** **boolean** equals(Mouse otherMouse)

{

**boolean** varietyCheck, weightCheck;

varietyCheck = variety.equals(otherMouse.variety);

weightCheck = (weight == otherMouse.weight);

**return** varietyCheck && weightCheck;

} // end equals

} // end class Mouse

1. Link breakpoint stops the program at a certain point this pauses the program so that it can be looked over otherwise it would run like normal.
2. CTRL+Shift+B
3. F5
4. Use Step Return to back out
5. Right click and select Inspect, it allows you to view the expression’s value.
6. Run to Line

PROJECT:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Book.java

\* <Adam Cox>

\*

\* This is the Superclass has 2 subclasses.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** Book

{

**double** manufacturCost = 0; // Book cost for creator

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** Book(**double** cost)

{

manufacturCost = cost;

} // end of constructor

} // end class Book

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* EBook.java

\* <Adam Cox>

\*

\* This is for EBooks, subclass of Book.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** EBook **extends** Book

{

String purchaseLink; // Site to get book.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** EBook(**double** price, String link)

{

**super**(price);

**this**.purchaseLink = link;

} // end constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **boolean** equals(EBook otherBook)

{

**return** otherBook.purchaseLink == **this**.purchaseLink

&& otherBook.manufacturCost == **this**.manufacturCost;

} // end equals

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **void** display()

{

System.***out***.printf("price = $%,.2f, url = \"%s\"\n",

**this**.manufacturCost\*.1, **this**.purchaseLink);

} // end display

} // end EBook

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PaperBook.java

\* <Adam Cox>

\*

\* This is for Paperbooks, subclass of Book.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** PaperBook **extends** Book

{

**double** bookWeight = 0; // Books Weight

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** PaperBook(**double** price, **double** weight)

{

**super**(price);

**this**.bookWeight = weight;

} // end constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **boolean** equals(PaperBook otherBook)

{

**return** otherBook.bookWeight == **this**.bookWeight

&& otherBook.manufacturCost == **this**.manufacturCost;

} // end equals

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **void** display()

{

System.***out***.printf("price = $%,.2f, weight = %,.1f",

**this**.manufacturCost\***this**.bookWeight, **this**.bookWeight);

} // end display

} // end PaperBook

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* BooksDriver.java

\* <Adam Cox>

\*

\* This is the driver for Books.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** BooksDriver

{

**public** **static** **void** main(String[] args)

{

EBook eBook1 =

**new** EBook(52.80, "www.books.com/TheHandmaidsTale.pdf");

EBook eBook2 =

**new** EBook(52.80, "www.books.com/TheHandmaidsTale.pdf");

**if** (eBook1.equals(eBook2))

{

eBook1.display();

}

PaperBook paper1 = **new** PaperBook(60, 2);

PaperBook paper2 = **new** PaperBook(60, 2.0);

**if** (paper1.equals(paper2))

{

paper1.display();

}

} // end main

} // end class BookDriver

price = $5.28, url = "www.books.com/TheHandmaidsTale.pdf"

price = $120.00, weight = 2.0