מבוא לתכנות מונחה עצמים

סטודנטית 1: דליה ויליאם

סטודנט 2: גיא רחמים

Book class:

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class Book
	{
		protected String name;
		protected int numberOfPages;
		protected String author;

		public Book()
		{
			setName("Generic " + getClass().getSimpleName());
			setNumberOfPages(100);
			setAuthor("Generic author");
		}

		public Book(String name, int numberOfPages, String author)
		{
			setName(name);
			setNumberOfPages(numberOfPages);
			setAuthor(author);
		}

		public String getName()
			{
				return name;
			}

		public void setName(String name)
			{
				this.name = name;
			}

		public int getNumberOfPages()
			{
				return numberOfPages;
			}

		public void setNumberOfPages(int numberOfPages)
			{
				if (numberOfPages<=0)
					{
						System.out.println("a book cant have fewer
than 1 pages. setting number of pages to 100.");
```

```
                                this.numberOfPages=100;
                        }
                else
                this.numberOfPages = numberOfPages;
        }

    public String getAuthor()
        {
                return author;
        }

    public void setAuthor(String author)
        {
                this.author = author;
        }

    public String summarize()
    {
        return "This is a " + getClass().getSimpleName()+".";
    }

    @Override
    public boolean equals(Object obj)
    {
        //is obj referencing a memory address?
        if (obj==null)
                return false;

        //are both objects referenceing the same address?
        if (this==obj)
                return true;

        //are both objects instantiated by the same class?
        if (this.getClass()!=obj.getClass())
                return false;

        //taking the reference held by obj and casting it to a Book type
object.
        Book other = (Book) obj;
        if (this.getName()!=other.getName())
                return false;

        //checking each individual field of class Book.
        if (this.getNumberOfPages()!=other.getNumberOfPages())
                return false;

        if (this.getAuthor()!=other.getAuthor())
                return false;

        //if we managed to get this far, the objects must be equal in
value.
        return true;
    }

    public String toString()
```

```
            {
                    return getClass().getSimpleName()+" name: " +getName() + ",
number of pages: " + getNumberOfPages()+ ", Author: " + getAuthor();
            }
        }
```

LibraryBookClass:

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class LibraryBook extends Book
	{
		protected int numberOfCopies;
		public LibraryBook()
		{
			super();
			setNumberOfCopies(10);
		}

		public LibraryBook(String name, int numberOfPages, String author,int
numberOfCopies)
		{
			super(name,numberOfPages,author);
			setNumberOfCopies(numberOfCopies);
		}

		public int getNumberOfCopies()
			{
				return numberOfCopies;
			}

		public void setNumberOfCopies(int numberOfCopies)
			{
				if(numberOfCopies<0)
					{
						System.out.println("Number of copies cant be
less than 0! setting to 1.");
						this.numberOfCopies=1;
					}
				else
					this.numberOfCopies = numberOfCopies;
			}



		@Override
		public String summarize()
		{
			return super.summarize()+" This book is for reading inside the
library only!";
		}

		public boolean borrow(int copiesToBorrow)
		{
			System.out.println("Sorry, a " +getClass().getSimpleName() + "
cannot be borrowed. Borrow failed.");
			return false;
		}

		public boolean returnBook (int copiesToReturn)
```

```
                                {
                                        System.out.println("You cant return a "+
getClass().getSimpleName()+" because you cant borrow it! return failed.");
                                        return false;
                                }
        }
```

ComicBook class:

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class ComicBook extends LibraryBook
	{

		public ComicBook()
		{
			super();
		}

		public ComicBook(String name, int numberOfPages, String authorName, int
numberOfCopies)
		{
			super(name,numberOfPages,authorName,numberOfCopies);
		}

		public int getNumberOfCopies()
		{
			return numberOfCopies;
		}

		public void setNumberOfCopies(int numberOfAvailableCopies)
		{
			if (numberOfAvailableCopies<0)
				{
					System.out.println("Minumum number of available
copies is 0. Setting the number to 1.");
					this.numberOfCopies=1;
				}
			this.numberOfCopies=numberOfAvailableCopies;
		}

		@Override
		public String summarize()
		{
			return "This is a generic " +getClass().getSimpleName() +". This
is for reading inside the library only!";
		}

		@Override
		public boolean equals(Object obj)
		{
			//are both objects referencing the same memory address?
			if (this==obj)
				return true;

			//checking for shared fields with super.
			if (!super.equals(obj))
				return false;

				ComicBook other = (ComicBook) obj; //casting obj to
ComicBook
```

```
                    if (this.getNumberOfCopies()!=other.getNumberOfCopies())
                        return false;
                    return true;

        }


}
```

ActionComicBook class

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class ActionComicBook extends ComicBook
	{
		protected int recommendedReadingAge;

		public ActionComicBook()
		{
			super();
			setRecommendedReadingAge(16);
		}

		public ActionComicBook(String name, int numberOfPages, String author,
int numberOfCopies, int recommendedReadingAge)
		{
			super (name,numberOfPages,author,numberOfCopies);
			setRecommendedReadingAge(recommendedReadingAge);
		}

		public int getRecommendedReadingAge()
			{
				return recommendedReadingAge;
			}

		public void setRecommendedReadingAge(int recommendedReadingAge)
			{
				if (recommendedReadingAge<1)
					{
						System.out.println("Recommended reading age
cant be be under 1. setting to 16.");
						this.recommendedReadingAge=16;
					}
				else
					this.recommendedReadingAge = recommendedReadingAge;
			}

		@Override
		public boolean borrow(int copiesToBorrow)
		{
			if (copiesToBorrow<=getNumberOfCopies())
				{
					setNumberOfCopies(getNumberOfCopies()-
copiesToBorrow);
					System.out.println("Borrow successful! Remaining
copies of " +getClass().getSimpleName()+": " + getNumberOfCopies());
					return true;
				}
			else
				{
					System.out.println("Sorry, there are not enough
available copies right now.");
					return false;
```

```
                        }
            }

            @Override
            public boolean returnBook(int amountToReturn)
            {
                  if (amountToReturn<1)
                        {
                              System.out.println("you need to have
"+getClass().getSimpleName()+"s in order to return them!");
                              return false;
                        }

                  setNumberOfCopies(getNumberOfCopies()+amountToReturn);
                  System.out.println(getClass().getSimpleName()+" Return
successful! number of available copies:"+ getNumberOfCopies());
                  return true;
            }
            @Override
            public String summarize()
            {
                  return "This " +getClass().getSimpleName()+ " is for ages greater
than " + getRecommendedReadingAge();
            }

            @Override
            public String toString()
            {
                  return super.toString() + ", Recommended reading age: "
+getRecommendedReadingAge();
            }

            @Override
            public boolean equals(Object obj)
            {
                  //are both objects referencing the same memory address?
                  if(this==obj)
                        return true;
                  if (!super.equals(obj))
                        return false;

                  ActionComicBook other = (ActionComicBook) obj;

                  if
(this.getRecommendedReadingAge()!=other.getRecommendedReadingAge())
                        return false;
                  return true;
            }

      }
```

CookBook class:

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class CookBook extends LibraryBook
	{
		protected int numberOfCopies;
		protected boolean isDamaged;

		public CookBook()
		{
			super();
			setNumberOfCopies(10);
			setIsDamaged(false);
		}

		public CookBook(String name, int numberOfPages, String author, int
numberOfCopies, boolean isDamaged)
		{
			super(name, numberOfPages, author,numberOfCopies);
			setIsDamaged(isDamaged);
		}


		public boolean getIsDamaged()
			{
				return isDamaged;
			}

		public void setIsDamaged(boolean isDamaged)
			{
				this.isDamaged = isDamaged;
			}

		public String toString()
		{
			String isDamagedString;
			if (isDamaged)
				isDamagedString = "the book is damaged.";
			else
				isDamagedString = "the book is intact!";
			return super.toString() + ", number of available copies: "
+numberOfCopies + " ," +isDamagedString;
		}

		@Override
		public boolean borrow(int copiesToBorrow)
		{
			if (copiesToBorrow<=getNumberOfCopies())
				{
					setNumberOfCopies(getNumberOfCopies()-
copiesToBorrow);
```

```
                                System.out.println("Borrow successful! Remaining
copies of " +getClass().getSimpleName()+": " + getNumberOfCopies());
                                return true;
                        }
                else
                        {
                                System.out.println("Sorry, there are not enough
available copies of " +getClass().getSimpleName()+"  right now.");
                                return false;
                        }
        }

        @Override
        public boolean returnBook(int amountToReturn)
        {
                if (amountToReturn<1)
                        {
                                System.out.println("you need to have books in order
to return them!");
                                return false;
                        }
                if (isDamaged)
                        {
                                fine();
                        }
                setNumberOfCopies(getNumberOfCopies()+amountToReturn);
                System.out.println(getClass().getSimpleName()+" Return
successful! number of available copies:"+ getNumberOfCopies());
                return true;
        }

        public void fine()
        {
                System.out.println("One of the " +getClass().getSimpleName()+"s
you returned is damaged!\nfine is 200NIS.");
        }

        @Override
        public String summarize()
        {
                return "This "+ getClass().getSimpleName()+" might contain non
cosher ingredients";
        }

        @Override
        public boolean equals(Object obj)
        {
                //are both objects referencing the same memory address?
                if (this==obj)
                        return true;
                //comparing shared super fields
                if (!super.equals(obj))
                        return false;

                //casting Object to CookBook
```

```
        CookBook other = (CookBook) obj;

        //checking values for non shared fields
        if (this.getIsDamaged()!=other.getIsDamaged())
            return false;
        return true;
    }

}
```

CosherCookBookClass:

```java
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package assignment4;

public class CosherCookBook extends CookBook
	{
		protected String supervision;

		public CosherCookBook()
		{
			super();
			setSupervision("the almighty flying spaggheti moster!");
		}

		public CosherCookBook(String name, int numberOfPages, String author, int
numberOfAvailableBooks, boolean isDamaged, String supervision)
		{
			super(name, numberOfPages, author, numberOfAvailableBooks,
isDamaged);
			setSupervision(supervision);
		}

		public String getSupervision()
			{
				return supervision;
			}

		public void setSupervision(String supervision)
			{
				this.supervision = supervision;
			}

		@Override
		public String summarize()
		{
			return "this " + getClass().getSimpleName()+"is supervised by " +
supervision;
		}

		@Override
		public boolean equals(Object obj)
		{
			//are both objects referencing the same memory address?
			if (this==obj)
				return true;

			//checking the shared fields with super.
			if (!super.equals(obj))
				return false;

			//casting to CosherCookBook
			CosherCookBook other = (CosherCookBook) obj;

			//checking for non shared fields.
```

```
            if (this.getSupervision()!=other.getSupervision())
                return false;
            return true;
        }
}
```

MainClass class:

```
//Dalya William 311529382 && Guy Rahamim Danino 313167686
package mainPackage;
import assignment4.*;

public class MainClass
      {

            public static void main(String[] args)
                  {
                        Book book = new Book("Generic book", 271, "Generic
author");
                        LibraryBook library = new LibraryBook("c-137", 200,
"Bob",30);
                        ComicBook comic = new ComicBook("The
awakening",314,"Anthony russo",15);
                        ActionComicBook wonderWoman = new ActionComicBook("Wonder
Woman", 628,"William Moulton Marston, H. G. Peter",7,18);
                        CookBook tasty = new CookBook("Tasty", 50, "Karin
Goren",4,true);
                        CosherCookBook joyOfCosher = new CosherCookBook("Joy of
Cosher", 30,"Rabbi Akiva",5,false,"maimon");

                        System.out.println("----------------------Calling borrow
methods------------------------");
                        System.out.println();
                        library.borrow(30);
                        comic.borrow(10);
                        wonderWoman.borrow(5);
                        tasty.borrow(7);
                        joyOfCosher.borrow(55);

                        System.out.println("\n\n------------------------calling
returnBook methods------------------------");
                        library.returnBook(50);
                        comic.returnBook(314);
                        wonderWoman.returnBook(1);
                        tasty.returnBook(5);
                        joyOfCosher.returnBook(271);

                        System.out.println("\n\n------------------------calling
toString methods------------------------");
                        System.out.println(book);
                        System.out.println(library);
                        System.out.println(comic);
                        System.out.println(wonderWoman);
                        System.out.println(tasty);
                        System.out.println(joyOfCosher);

                        System.out.println("\n\n------------------------calling
summarize methods------------------------");
                        System.out.println(book.summarize());
                        System.out.println(library.summarize());
                        System.out.println(comic.summarize());
```

```
        System.out.println(wonderWoman.summarize());
        System.out.println(tasty.summarize());
        System.out.println(joyOfCosher.summarize());
    }
}
```

Output example:

```
----------------------Calling borrow methods------------------------

Sorry, a LibraryBook cannot be borrowed. Borrow failed.
Sorry, a ComicBook cannot be borrowed. Borrow failed.
Borrow successful! Remaining copies of ActionComicBook: 2
Sorry, there are not enough available copies of CookBook  right now.
Sorry, there are not enough available copies of CosherCookBook  right now.


------------------------calling returnBook methods------------------------
You cant return a LibraryBook because you cant borrow it! return failed.
You cant return a ComicBook because you cant borrow it! return failed.
ActionComicBook Return successful! number of available copies:3
One of the CookBooks you returned is damaged!
fine is 200NIS.
CookBook Return successful! number of available copies:9
CosherCookBook Return successful! number of available copies:276


------------------------calling toString methods------------------------
Book name: Generic book, number of pages: 271, Author: Generic author
LibraryBook name: c-137, number of pages: 200, Author: Bob
ComicBook name: The awakening, number of pages: 314, Author: Anthony russo
ActionComicBook name: Wonder Woman, number of pages: 628, Author: William Moulton Marston, H. G. Peter, Recommended reading age: 18
CookBook name: Tasty, number of pages: 50, Author: Karin Goren, number of available copies: 0 ,the book is damaged.
CosherCookBook name: Joy of Cosher, number of pages: 30, Author: Rabbi Akiva, number of available copies: 0 ,the book is intact!


------------------------calling summarize methods------------------------
This is a Book.
This is a LibraryBook. This book is for reading inside the library only!
This is a generic ComicBook. This is for reading inside the library only!
This ActionComicBook is for ages greater than 18
This CookBook might contain non cosher ingredients
this CosherCookBookis supervised by maimon
```