CS 1150 Design Notebook Required Sections

Step 1: Problem Statement

In this assignment we will create a polymorphic array and read lines from a given file to fill array with objects and from these objects will display name, type, and motto each object has and at the end create a new object to call 2 methods that take in the polymorphic array.

Step 2: Understandings

- What I Know:
  - Polymorphic Arrays
  - For Loop
  - Objects mostly
- What I Don't Know:
  - It's been a while since I did objects
  - File reading, not enough practice with it

Step 3: Pseudocode

Main:

- Open giving "actors" file
  - File filename = new File(Actors.txt)
- Create a polymorphic array using the number given in file for array size
  - Use nextInt()
- File the array with objects given from file
  - Use Switch Statement
- For loop through the array to display each object's name, type, and motto
- Create a new movie object
- Call selectCast with array
- Call printMovieDetails

Step 4: Lesson Learned

I did not get stuck anywhere during this assignment, it was a good refresher to objects and subclasses and more practice with file reading. I did learn however, that you must use Java's built in File class to read a file even if I already had the file in my project folder.

Step 5: Code

```
//package cs1450;

/*
Isaiah Hoffer
CS1450 (M/W)
2/5/25
Assignment 2
This assignment will use a file to create objects that are children of the Actor class
and use polymorphism to create an array of objects and display certain information about
the objects and will create a new array with only certain objects in them and display those objects.
*/
```

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Scanner;

public class HofferIsaiahAssignment2 {

	public static void main(String[] args) throws IOException {

		//File Name Constant
		String FILE_NAME_STR = "Actors.txt";

		//Variable For Actors File
		File fileName = new File(FILE_NAME_STR);


		//Scanner To Read File
		Scanner readFile = new Scanner(fileName);
		//Creating Polymorphic Array
		Actor[] actorObjArray = new Actor[readFile.nextInt()];

		for(int i = 0; i < actorObjArray.length; i++) {

			String actorType = readFile.next();
			String actorName = readFile.nextLine();

			//Finding Object Type
			switch(actorType) {
				case "Hero":
					actorObjArray[i] = new Hero(actorName);
					break;
				case "Villain":
					actorObjArray[i] = new Villain(actorName);
					break;
				case "Monster":
					actorObjArray[i] = new Monster(actorName);
					break;
				case "Droid":
					actorObjArray[i] = new Droid(actorName);
					break;


			}//Switch

		}//For

		//Closing File
		readFile.close();

		//Displaying Each Object's Name, Type, and Motto
			//Pretext
```

```
                System.out.printf("-----------------------------------------------------------------------------------------
\n"
                                + "Actor Name\t\tType\t\t\t\tMotto To Live By\n"
                                + "-----------------------------------------------------------------------------------------
\n");

                for(int i = 0; i < actorObjArray.length; i++) {

                        System.out.printf("%-
10s\t\t%s\t\t\t%s\n",actorObjArray[i].getName().trim(),actorObjArray[i].getType(),actorObjArray[i].
motto());

                }//For

                //Creating Movie Object
                Movie movieObj = new Movie();

                //Calling selectCast Method
                movieObj.selectCast(actorObjArray);

                //Calling printMovieDetails Method
                movieObj.printMovieDetails();

        }//main
}//Class


//Parent Class- SubClasses: Hero, Villian, Monster, Droid
//Returns actor's name, type, and motto
class Actor {

        //Class Data Fields
        private String type;
        private String name;

        public Actor() {}//Actor, No Args

        //Constructor
        public Actor(String name, String type) {

                //Setting Values
                this.name = name;
                this.type = type;

        }//Actor Con.

        //Getter to Get Name
        public String getName() {

                return name;
        }//getName
```

```
	//Getter to Get Type
	public String getType() {

		return type;
	}//getType

	//Returns Actors Motto
	public String motto() {

		return "Woopsie";
	}

}//Actor Class


//Hero Class
class Hero extends Actor {

	//Constructor
	public Hero(String name) {

		//Sending Info To Parent(Actors)
		super(name,"Hero");

	}//Hero Cons.

	@Override
	public String motto() {

		return "To the rescue! KAPOW!! BAM!! POW!!";
	}

}//Hero Class

//Hero Class
class Villain extends Actor {

	//Constructor
	public Villain(String name) {

		//Sending Info To Parent(Actors)
		super(name,"Villain");

	}//Villain Cons.

	@Override
	public String motto() {

		return "You'll never stop me! Haaaaaa!";
	}
```

```
}//Villain Class


//Monster Class
class Monster extends Actor {

	//Constructor
	public Monster(String name) {

		//Sending Info To Parent(Actors)
		super(name,"Monster");

	}//Monster Cons.

	@Override
	public String motto() {

		return "RRAAAUUGGHH GRROWR!!!";
	}

}//Monster Class

//Hero Class
class Droid extends Actor {

	//Constructor
	public Droid(String name) {

		//Sending Info To Parent(Actors)
		super(name,"Droid");

	}//Droid Cons.

	@Override
	public String motto() {

		return "Beep Beep Bloop Boop Beep!";
	}

}//Droid Class


class Movie {

	//Setting Class' Private Data
	private int numHeroes;
	private int numVillains;
	private Actor[] actorsInMovie;

	//Fill actorsInMovie Array with Only Villains and Heros
```

```
	public void selectCast(Actor[] cast) {

		//Finding Heros and Villains Objects
		for(int i = 0; i < cast.length; i++) {

			if(cast[i] instanceof Hero) {
				numHeroes++;
			}//If
			else if(cast[i] instanceof Villain) {

				numVillains++;

			}//Else If

		}//For

		actorsInMovie = new Actor[numHeroes + numVillains];

		//Counter For Indexing
		int movieIndex = 0;
		//Setting actorsInMovie Values  (Heros and Vliiains)
		for(int i = 0; i < cast.length; i++) {

			if(cast[i] instanceof Hero ||  cast[i] instanceof Villain) {

				actorsInMovie[movieIndex] = cast[i];
				movieIndex++;
			}//If
		}//For

	}//selectCast

	//Displays actorsInMovie and Certain Info
	public void printMovieDetails() {

		//Pretext
		System.out.printf("--------------------------------\n"
				+ "CS1450 Heroes and Villain Movie\n"
				+ "--------------------------------");

		//Displaying Amount of Heroes and Villains
		System.out.printf("\nNumber Of Heroes:   %d\n"
				+ "Number Of Villains: %d\n\n", numHeroes, numVillains);


		//Displaying Heros and Villians Movie Cast
		for(int i = 0; i < actorsInMovie.length; i++) {

			System.out.printf("%s\t\t---%s\n",actorsInMovie[i].getType(),
actorsInMovie[i].getName().trim());
```

```
            }//For


        }//printMovieDetails

}//Movie Class
```