

Create an interface named `Payable` with the following method

- `getPaymentAmount() : returns double`

Create a class `Employee` that will implement the interface `Payable` but will not override the method `getPaymentAmount()`. `Employee` class must have the following form:

- `firstName: String`
- `lastName: String`
- `socialSecurityNumber: String`
- A constructor for all the above mentioned variables
- Getter methods for all the above mentioned variables.
- A `toString()` method to return a string concatenating all the variables.

Create a class `SalariedEmployee` that will extend the class `Employee`. The class must have the following methods and variables:

- `weeklySalary: double`
- an appropriate constructor. However the constructor must check to see if the value that is being assigned to the `weeklySalary` is positive, otherwise it must throw an `IllegalArgumentException`.
- Getter method for `weeklySalary`
- Setter method for `weeklySalary`. Again throw an exception of type `IllegalArgumentException` is the new `weeklySalary` is negative.
- `getPaymentAmount() : double`. This method must return `weeklySalary`
- A `toString` method that will add an extra string "salaried employee" with `Employee` class's `toString` method

Create a class `Invoice` that must implement the `Payable` interface with the following form:

- `partNumber: String`
- `partDescription: String`
- `quantity: int`
- `price: double`

- Appropriate constructor for all of the four variables. The constructor must throw `IllegalArgumentException`, if quantity or price is negative.
- Getter and setters for all four variables. However, the setter methods for price and quantity must ensure that they are positive otherwise throw an `IllegalArgumentException`.
- A `toString` method that will return all four variables as a `String` separated by space.

To test everything from the `Main` class, create an array of type `Payable` of length 4. Assign two objects of type `SalariedEmployee` and `Invoice`, each to the array. Use dynamic method dispatching to call `toString` and `getPaymentAmount` and print the result on the console, via `Payable` interface reference.