**Documentation for Online Shopping Site-Project 2**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
\* **Programmer**: Sravya Chava   
\*   
\* **Course**:  CSCI 4391.05   
\*   
\* **Date**:  02/25/2019

\*   
\* **Assignment**:  Program #2: Online Shopping Site  
\*   
\* **Environment**:  Java with IntelliJ IDE  
\*   
\* **Files Included**: WebUser.java, ShoppingCart.java, Product.java, LineItem.java, Payment.java, \* Order.java, Product.java, OrderStatus.java, Account.java, Userstate.java, Customer.java, \* \* \* \* User.java, Driver.java  
\*   
\* **Purpose**: Implementing online Shopping site classes using cloneable, Generics, Comparable, \* Comparator, Bounded types and wildcards.  
\*   
\* **Input**: Not Available  
\*   
\* **Preconditions/Assumptions**:  Not available  
\*   
\* **Output**:  Gives a cloned customer object using cloneable, sorted Orders using comparable, sorting \* \*\*productsusing a name comparator, Users extending web user using bounded types,   
\*   
\* **Postconditions**:  Not available  
\*   
\* **Algorithm**:

\* Construct all the required classes using Generics.

\* Declare all the parameters with the generic data types including the parameters having \* \* \* \* reference variable of classes.  
\*      Generate getters and setters to all the private variables.

\* Generate the constructor and to String methods in all the classes.  
\*      Customer class implements the cloneable interface and overrides clone method to return \* customer object.

\* In Driver class customer object is created using setters.

\* Also, new customer object is created by cloning the old object using .clone.

\* Now print all the old object parameters using getters.

\* Comparator is imported to Order class.

\* Order class implements Comparable.

\* compareTo method of Comparable is overridden to compare all the orders based on order \* number.

\* In Driver Class order Objects array(orderArray) is created with different order objects.

\* orderArray is sorted using Arrays.sort(orderarray)

\* Comparator is imported to Product class.

\* Product class has a Name comparator in which compare method is overridden.

\* In Driver class product objects array is created with different products.

\* productArray is sorted using NameComparator using Arrays.sort(productArray, \* \* \* \* \* \* NameComparator).

\* Create a WebUser class.

\* User class extends WebUser And Student User extends WebUser, StaffUser extends \* WebUser.

\* Using bounded type with generics, an class unknown type can extend WebUser.

\* We can then the method in all the classes by passing respective objects in place of unknown \* type.

\* Upper wild cards are implemented by Account extending Parentaccount

\* A method is creating in which the list of objects supers the ParentAccount, method can \* accept only ParentAccount and its subtype Account.

\* Lower wild cards are implemented by CashPayment extending Payment class

\* A method is creating in which the list of objects extends the Payment, method can \* accept only CashPayment and its supertype Payment class.

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