Martin Hynes
16390836

Sales Employee.java

```
//Create abstract SalesEmployee class, so there can be no instance of
it.
public abstract class SalesEmployee{//open class


	//Instance Variables:


	//private name and PPS. String for PPS because of letters in it.
	private String name;
	private String pps;
	//protected sales and commission variables, so they can be easily
accessed in subclasses.
	protected double sales;
	protected double commission;


	//Abstract Methods


	//abstract calculateCommission method, has to be overwritten in
subclasses.
	abstract double calculateCommission();


	//Constructors


	//General constructor for no inputs. Set all values to 0 or
String equivelent.
	public SalesEmployee(){//open constructor
		this.name = "unassigned";
		this.pps = "unassigned";
		this.sales = 0;
		this.commission = 0;
	}//close constructor

	//Overload constructor. Take Name, PPS, and Sales as input.
Calculate commission from given sales.
	public SalesEmployee(String Name,String PPS, double Sales){//open
overload constructor
		this.name = Name;
		this.pps = PPS;
		this.sales = Sales;
		this.commission = this.calculateCommission();
	}//close overload constructor
```

```
	//Accessor methods


	//Getter and Setter for name
	public String getName(){//open Name getter
		return this.name;
	}//close name getter

	public void setName(String Name){//open name setter
		this.name = Name;
	}//close name setter

	//Getter and Setter for PPS
	public String getPPS(){//open pps getter
		return this.pps;
	}//close pps getter

	public void setPPS(String PPS){//open pps setter
		this.pps = PPS;
	}//close pps setter

	//Getter and Setter for Sales
	public double getSales(){//open sales getter
		return this.sales;
	}//close sales getter
	//Changing sales automatically updates the commission value.
	public void setSales(double Sales){//open sales setter
		this.sales = Sales;
		this.commission = this.calculateCommission();
	}//close sales setter
	//Get commission method
	public double getCommission(){//open commission getter
		return this.commission;
	}//close comission getter

	//Overwritten toString to summarise stored valiables.
	public String toString(){//open toString overwrite method
		return "Sales Employee. Name: " + this.name + " PPS: " +
this.pps + " Sales: " + this.sales + " Commission: " +
this.commission;
	}//close toString overwrite method
}//close class
```

SalesAgent.java

```
//SalesAgent class, subclass of Sales Employee
public class SalesAgent extends SalesEmployee{//open class


	//Constructors
```

```
	//Use superclass constructor for generic constructor
	public SalesAgent(){//open constructor
		super();
	}//close constructor

	//Set instance variables to given values.
	public SalesAgent(String Name, String PPS, double Sales){//open
overload constructor
		super();
		this.setName(Name);
		this.setPPS(PPS);
		this.sales = Sales;
		//use calculateCommission method to set commission variable
		this.commission = this.calculateCommission();
	}//close overload constructor


	//Methods


	//CalculateCommission uses 10% commission for Sales Agents
	public double calculateCommission(){//open calculateCommission
method
		return this.sales * 0.10;
	}//close calculateCommission method

	//Overwrite toString method
	public String toString(){//open toString override
		return "Sales Agent. Name: "+this.getName()+" PPS: " +
this.getPPS() + " Sales: " + this.getSales() + " Commission: " +
this.getCommission();
	}//close toString override
}//close class
```

SalesPerson.java

```
//SalesPerson class, subclass of Sales Employee
public class SalesPerson extends SalesEmployee{//open class


	//Constructors


	//Use constructor from superclass for generic constructor
	public SalesPerson(){//open Constructor
		super();
	}//close Constructor
```

```
     //Set instance variables to inputted arguments in overload
constructor
     public SalesPerson(String Name, String PPS, double Sales){//open
Overload Constructor
          super();
          this.setName(Name);
          this.setPPS(PPS);
          this.sales = Sales;
          //Set commission using the Sales value
          this.commission = this.calculateCommission();
     }//close Overload Constructor


     //Methods


     //CalculateCommission used 15% commission for SalesPerson
     public double calculateCommission(){//open calculateCommission
method
          return this.sales * 0.15;
     }//close calculateCommission method

     //Overwrite toString
     public String toString(){//open toString overwrite method
          return "Sales Person. Name: "+this.getName()+" PPS: " +
this.getPPS() + " Sales: " + this.getSales() + " Commission: " +
this.getCommission();
     }//close toString overwrite method
}//close class
```

SalesEmployeeTest.java

```
//Import java util for ArrayList and Iterator
import java.util.*;

//Sales Employee Tester Class
public class SalesEmployeeTest{//open class
     public static void main(String[] args){//open main method

          //Create ArrayList for SalesEmployee objects, called
EmployeeList.
          ArrayList<SalesEmployee> EmployeeList = new
ArrayList<SalesEmployee>();
          //Populate ArrayList with 5 Employees, mix of SalesPerson
and SalesAgent object types.
          //No SalesEmployee objects because it is an abstract class.
          EmployeeList.add(new SalesPerson("Adam","123456A",5000));
          EmployeeList.add(new SalesAgent("Bob","123456B",10000));
          EmployeeList.add(new
SalesPerson("Charlie","123456C",15000));
```

```
            EmployeeList.add(new SalesAgent("Dan","123456D",20000));
            EmployeeList.add(new SalesPerson("Ellie","123456E",25000));
            //Create iterator to traverse the ArrayList.
            Iterator itr = EmployeeList.iterator();

            //While loop for displaying values of objects.
            while(itr.hasNext()){//open while loop
                  //Defining Emp to be the next object in the list each
iteration
                  SalesEmployee Emp = (SalesEmployee)itr.next();
                  //pring out Emp's Name, PPS, Sales, and Commission
                  System.out.println("Name: " + Emp.getName() );
                  System.out.println("PPS: " + Emp.getPPS());
                  System.out.println("Sales: " + Emp.getSales());
                  System.out.println("Commission: "+
Emp.getCommission());
                  System.out.println();
            }//close while loop
      }//close main method
}//close class
```

```
Name: Adam
PPS: 123456A
Sales: 5000.0
Commission: 750.0

Name: Bob
PPS: 123456B
Sales: 10000.0
Commission: 1000.0

Name: Charlie
PPS: 123456C
Sales: 15000.0
Commission: 2250.0

Name: Dan
PPS: 123456D
Sales: 20000.0
Commission: 2000.0

Name: Ellie
PPS: 123456E
Sales: 25000.0
Commission: 3750.0
```