A blue and white logo

Description automatically generated **San Francisco Bay University**

**EE616 Functional Verification with System Verilog**

**Quiz #2**

1. Package-delivery services, such as , and , offer a number of different shipping options, each with specific costs associated. Create an inheritance hierarchy to represent various types of packages. Use class *Package* as the base class of the hierarchy, then include classes *TwoDayPackage* and *OvernightPackage* that derive from *Package*.

Base class *Package* should include data members representing the name, address, city, state and ZIP code for both the sender and the recipient of the package, in addition to data members that store the weight (in ounces) and cost per ounce to ship the package. *Package*’s constructor should initialize these data members. Ensure that the weight and cost per ounce contain positive values. *Package* should provide a *public* member function *calculateCost* that returns a *real* indicating the cost associated with shipping the package. *Package’s calculateCost* function should determine the cost by multiplying the weight by the cost per ounce. Derived class *TwoDayPackage* should inherit the functionality of base class *Package*, but also include a data member that represents a flat fee that the shipping company charges for two-day-delivery service. *TwoDayPackage’s* constructor should receive a value to initialize this data member. *TwoDayPackage* should redefine member function *calculateCost* so that it computes the shipping cost by adding the flat fee to the weight-based cost calculated by base class *Package’s calculateCost* function. Class *OvernightPackage* should inherit directly from class *Package* and contain an additional data member representing an additional fee per ounce charged for overnight-delivery service. *OvernightPackage* should redefine member function *calculateCost* so that it adds the additional fee per ounce to the standard cost per ounce before calculating the shipping cost. Write a *program* structure that creates objects of each type of *Package* and tests member function *calculateCost*.

Program

module PDS;

class Package;

string senderName;

string senderAddress;

string senderCity;

string senderState;

int senderZIP;

string recipientName;

string recipientAddress;

string recipientCity;

string recipientState;

int recipientZIP;

real weight; // in ounces

real shipment\_costPerOunce;

// Constructor for Package

function new(string senderName, string senderAddress, string senderCity, string senderState, int senderZIP,

string recipientName, string recipientAddress, string recipientCity, string recipientState,

int recipientZIP, real weight, real shipment\_costPerOunce);

this.senderName = senderName;

this.senderAddress = senderAddress;

this.senderCity = senderCity;

this.senderState = senderState;

this.senderZIP = senderZIP;

this.recipientName = recipientName;

this.recipientAddress = recipientAddress;

this.recipientCity = recipientCity;

this.recipientState = recipientState;

this.recipientZIP = recipientZIP;

this.weight = (weight > 0) ? weight : 0;

this.shipment\_costPerOunce = (shipment\_costPerOunce > 0) ? shipment\_costPerOunce : 0;

endfunction

// Member function to calculate the cost for shipping the package

virtual function real calculateCost();

return weight \* shipment\_costPerOunce;

endfunction

endclass

class TwoDayPackage extends Package;

real flatFee;

// Constructor for TwoDayPackage

function new(string senderName, string senderAddress, string senderCity, string senderState, int senderZIP,

string recipientName, string recipientAddress, string recipientCity, string recipientState,

int recipientZIP, real weight, real shipment\_costPerOunce, real flatFee);

super.new(senderName, senderAddress, senderCity, senderState, senderZIP,

recipientName, recipientAddress, recipientCity, recipientState, recipientZIP,

weight, shipment\_costPerOunce);

this.flatFee = (flatFee > 0) ? flatFee : 0;

endfunction

// Redefining the member function to calculate the cost for two-day delivery

virtual function real calculateCost();

return super.calculateCost() + flatFee;

endfunction

endclass

class OvernightPackage extends Package;

real additionalFeePerOunce;

// Constructor for OvernightPackage

function new(string senderName, string senderAddress, string senderCity, string senderState, int senderZIP,

string recipientName, string recipientAddress, string recipientCity, string recipientState,

int recipientZIP, real weight, real shipment\_costPerOunce, real additionalFeePerOunce);

super.new(senderName, senderAddress, senderCity, senderState, senderZIP,

recipientName, recipientAddress, recipientCity, recipientState, recipientZIP,

weight, shipment\_costPerOunce);

this.additionalFeePerOunce = (additionalFeePerOunce > 0) ? additionalFeePerOunce : 0;

endfunction

// Redefining the member function to calculate the cost for overnight delivery

virtual function real calculateCost();

return super.calculateCost() + (weight \* additionalFeePerOunce);

endfunction

endclass

Package pkg;

TwoDayPackage twoDayPkg;

OvernightPackage overnightPkg;

initial begin

pkg = new("Hossian", "100 Oak St", "Cityville", "CA", 54321,

"Jane Doe", "200 Maple Ave", "Townsville", "NY", 98765,

12.5, 1.2);

$display("Cost for Package: $%0.2f", pkg.calculateCost());

twoDayPkg = new("Melvin", "300 Elm Rd", "Villageville", "TX", 12345,

"Divine", "400 Pine Ln", "Orinda", "IL", 67890,

10.0, 1.0, 15.0);

$display("Cost for Two-Day Package: $%0.2f", twoDayPkg.calculateCost());

overnightPkg = new("Aaron", "Richmond", "Countryside", "FL", 98765,

"Simon", "600 Cedar Pl", "Countryside", "OH", 54321,

8.0, 1.5, 3.5);

$display("Cost for Overnight Package: $%0.2f", overnightPkg.calculateCost());

end

endmodule

**running result**

A screenshot of a computer screen

Description automatically generated