|  |  |
| --- | --- |
| Student Name | Asadullah |
| Roll Number | 21SW036 |
| Section # | 03 |
| Lab # | 02 |

**Task#01**

Question statement

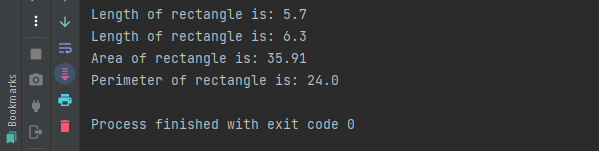
Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and lessthan20.0.

# Q1.Java

**Code:**

class Rectangle{  
  
 public float length = 1, width= 1;  
  
 public float perimeter(){  
 return 2 \* (this.length + this.width);  
 }  
  
 public float area(){  
 return (this.length \* this.width);  
 }  
  
 public void setLength(float length) {  
 if (length>0.0 && length < 20.0) {  
 this.length = length;  
 } else {  
 System.*out*.println("Please enter length between 0 and 20");  
 }  
 }  
  
 public float getLength() {  
 return this.length;  
 }  
  
 public void setWidth(float width) {  
 if (width>0.0 && width < 20.0) {  
 this.width = width;  
 } else {  
 System.*out*.println("Please enter width between 0 and 20");  
 }  
 }  
  
 public float getWidth() {  
 return this.width;  
 }  
  
} // end of class Rectangle  
  
public class Q1 {  
  
 public static void main(String[] args) {  
  
 Rectangle rectangle = new Rectangle();  
  
 rectangle.setLength(5.7f);  
 System.*out*.println("Length of rectangle is: "+rectangle.getLength());  
 rectangle.setWidth(6.3f);  
 System.*out*.println("Length of rectangle is: "+rectangle.getWidth());  
 System.*out*.println("Area of rectangle is: "+rectangle.area());  
 System.*out*.println("Perimeter of rectangle is: "+rectangle.perimeter());  
  
 } // end of main() method  
} // end of program

**Output:**

****

**Task#02**

Question statement

Create a class called Employee that includes three pieces of information as data members—a first name (type char\*), a last name (type string) and a monthly salary (type int). Your class should have a setter function that initializes the three data members. Provide a getter function for each data member. If the monthly salary is not positive, set it to 0.Write a test program that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10 percent raise and display each Employee’s yearly salary again. Identify and add any other related functions to achieve the said goal.

# Q2.Java

**Code:**

class Employee {  
 String firstName;  
 String lastName;  
 int salary;  
  
 public void setDetails(String firstName, String lastName, int salary) {  
 if (salary<0){  
 this.salary = 0;  
 } else {  
 this.salary = salary;  
 }  
 this.firstName = firstName;  
 this.lastName = lastName;  
 }  
 public int getSalary(){  
 return this.salary;  
 }  
 public String getFirstName(){  
 return this.firstName;  
 }  
 public String getLastName(){  
 return this.lastName;  
 }  
  
} // end of class Employee  
  
public class Q2 {  
  
  
 public static void main(String[] args) {  
  
 Employee employee1 = new Employee();  
 Employee employee2 = new Employee();  
  
 employee1.setDetails("John", "Doe", 55000);  
 employee2.setDetails("John", "Dalton", 25000);  
  
 int yearlySalary1 = employee1.getSalary() \* 12;  
 int yearlySalary2 = employee2.getSalary() \* 12;  
  
 System.*out*.println("Yearly salary of employee 1 is "+yearlySalary1);  
 System.*out*.println("Yearly salary of employee 2 is "+yearlySalary2);  
  
 yearlySalary1 += (yearlySalary1 \* 10)/100;  
 yearlySalary2+= (yearlySalary2 \* 10)/100;  
  
 System.*out*.println("Yearly salary of employee 1 after raise of 10% is "+yearlySalary1);  
 System.*out*.println("Yearly salary of employee 2 after raise of 10% is "+yearlySalary2);  
  
  
 System.*out*.print("Name of employee 1 is "+employee1.getFirstName()+" "+employee1.getLastName());  
 System.*out*.println();  
  
 System.*out*.print("Name of employee 2 is "+employee2.getFirstName()+" "+employee2.getLastName());  
 System.*out*.println();  
  
 Employee e = new Employee();  
 e.setDetails("B", "c", -5);  
 System.*out*.println(e.getSalary());  
  
  
 } // end of main() method  
} // end of program

**Output:Text

Description automatically generated**

**Task#03**

Question statement

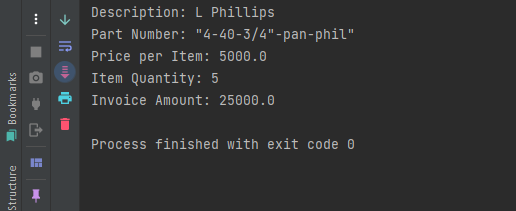
Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members—a part number (type string), a part description (type string), a quantity of the item being purchased(type int) and a price per item(type float).Your class should have a functions that initializes the four data members. Provide a get function for each data member. In addition, provide a member function named get Invoice Amount that calculates the invoice amount(i.e. multiplies the quantity by the price per item),then returns the amount as a float value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.Writeatestprogramthat demonstrates class Invoice’s capabilities.

# Q3.Java

**Code:**

class Invoice{  
 String partNumber, description;  
 int itemQuantity;  
 float pricePerItem;  
  
 public void setPartNumber(String partNumber) {  
 this.partNumber = partNumber;  
 }  
  
 public void setDescription(String description) {  
 this.description = description;  
 }  
  
 public void setItemQuantity(int itemQuantity) {  
 if (itemQuantity < 0){  
 itemQuantity = 0;  
 }  
 this.itemQuantity = itemQuantity;  
 }  
  
 public void setPricePeritemQuantity(float pricePerItem) {  
 if (pricePerItem < 0.0f){  
 pricePerItem = 0.0f;  
 }  
 this.pricePerItem = pricePerItem;  
 }  
  
 public String getPartNumber() {  
 return partNumber;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public int getItemQuantity() {  
 return itemQuantity;  
 }  
  
 public float getPricePerItem() {  
 return pricePerItem;  
 }  
  
 public float getInvoiceAmount(){  
 return (this.itemQuantity \* this.pricePerItem);  
 }  
  
} // end of class Invoice  
  
public class Q3 {  
 public static void main(String[] args) {  
  
 Invoice invoice = new Invoice();  
  
 invoice.setDescription("L Phillips");  
 invoice.setItemQuantity(5);  
 invoice.setPricePeritemQuantity(5000.00f);  
 invoice.setPartNumber("\"4-40-3/4\"-pan-phil\"");  
  
 System.*out*.println("Description: "+invoice.getDescription());  
 System.*out*.println("Part Number: "+invoice.getPartNumber());  
 System.*out*.println("Price per Item: "+invoice.getPricePerItem());  
 System.*out*.println("Item Quantity: "+invoice.getItemQuantity());  
 System.*out*.println("Invoice Amount: "+invoice.getInvoiceAmount());  
  
 } // end of main() method  
} // end of program

**Output:**

****

**Task#04**

Question statement

Write Java code to represent a hitting game. The details are as follows:

This game is being played between two teams(i.e. your team and the enemy team).The total number of players in your team is randomly generated and stored accordingly. The function generates a pair of numbers and matches each pair. If the numbers get matched, the following message is displayed:

“Enemy got hit by your team!”

Otherwise, the following message Is displayed:

“You got hit by the enemy team!”

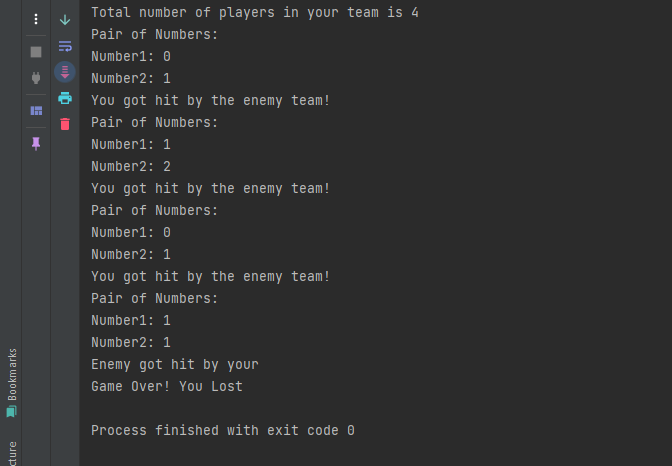
The number of hits should be equal to the number of players in your team. The program should tell the final result of your team by counting the hits of both the teams.

# Q4.Java

**Code:**

public class Q4 {  
  
 public static void Hits(){  
  
 // Algorithms for Random number generation between minValue and maxValue is  
// (int)(Math.random() \* (maxValue - minValue) + 1) + minValue;  
 int playerHits = 0, enemyHits = 0, totalHits = 1;  
 int numberOfPlayers = (int)(Math.*random*() \* (10 - 1) + 1) + 1; // Generate random number between 1 and 10  
 System.*out*.println("Total number of players in your team is "+numberOfPlayers);  
  
 while (totalHits<=numberOfPlayers){  
 System.*out*.println("Pair of Numbers:");  
 int num1 = (int)(Math.*random*() \* 3); // Generate random number between 0 and 3  
 int num2 = (int)(Math.*random*() \* 3); // Generate random number between 0 and 3  
 System.*out*.println("Number1: "+num1);  
 System.*out*.println("Number2: "+num2);  
  
 if (num1==num2){  
 System.*out*.println("Enemy got hit by your");  
 playerHits++;  
 } else {  
 System.*out*.println("You got hit by the enemy team!");  
 enemyHits++;  
 }  
 totalHits++;  
 } // end of while loop  
  
 if (playerHits>enemyHits){  
 System.*out*.println("Game Over! You Won");  
 } else if (playerHits<enemyHits){  
 System.*out*.println("Game Over! You Lost");  
 } else if (playerHits==enemyHits){  
 System.*out*.println("Game Over! It's a tie");  
 }  
  
 } // end of Hits() method  
  
  
 public static void main(String[] args) {  
  
 *Hits*();  
  
 } // end of main() method  
} // end of program

**Output:**

****

**Task#05**

Question statement

MyJava Coffee Outlet runs a catalog business. It sells only one type of coffee beans. The company sells the coffee in 2- lb bags only and the price of a single 2-lb bag is $5.50 when a customer places an order, the company ships the order in boxes. The boxes come in 3 sizes with 3 different costs: Capacity Large Box 20 Bags Medium Box 10 Bags Small Box 5 Bags Cost $1.80 $1.00 $0.60 The order is shipped using the least number of boxes. For example, the order of 52 bags will be shipped in 2 boxes: 2large boxes, 1medium and 1 small. Develop an application that computes the total cost of an order. Number of Bags Ordered: 52The Cost of Order: $ 286.00 Boxes Used: 2 Large - $3.60 1 Medium - $1.001 Small - $0.60 Your total cost is: $ 291.2

# Q5.Java

**Code:**

class Outlet{  
  
 float cost = 5.50f, largeBoxCost = 1.80f, mediumBoxCost = 1.00f, smallBoxCost = 0.60f;  
 int order, mediumBoxes, smallBoxes, largeBoxes;  
  
 public void setOrder(int order){  
 this.order = order;  
 }  
  
 public int getOrder() {  
 return order;  
 }  
  
 public float getOrderCost(){  
 return this.order \* cost;  
 }  
 public float getLargeBoxesCost(){  
 largeBoxes = this.order / 20;  
 System.*out*.print(largeBoxes + " Large - $");  
 return largeBoxes \* largeBoxCost;  
 }  
  
 public float getMediumBoxesCost(){  
 int remainingBoxes = this.order % 20;  
 mediumBoxes = remainingBoxes / 10;  
 System.*out*.print(mediumBoxes + " Medium - $");  
 return mediumBoxes \* mediumBoxCost;  
 }  
  
 public float getSmallBoxesCost(){  
 smallBoxes = mediumBoxes % 10;  
 System.*out*.print(smallBoxes + " Small - $");  
 return smallBoxes \* smallBoxCost;  
 }  
  
 public float totalCost(){  
 return (this.order \* cost) + (largeBoxes \* largeBoxCost) + (mediumBoxes \* mediumBoxCost) + (smallBoxes \* smallBoxCost);  
 }  
  
} // end of class Outlet  
  
public class Q5 {  
 public static void main(String[] args) {  
  
 Outlet outlet = new Outlet();  
 outlet.setOrder(52);  
 System.*out*.println("Number of Bags Ordered: "+outlet.getOrder());  
 System.*out*.println("The Cost of Order: $"+outlet.getOrderCost());  
 System.*out*.println("Used: ");  
 System.*out*.println(outlet.getLargeBoxesCost());  
 System.*out*.println(outlet.getMediumBoxesCost());  
 System.*out*.println(outlet.getSmallBoxesCost());  
 System.*out*.println("Your total cost is "+outlet.totalCost());  
 }  
}

**Output:**

**Text

Description automatically generated**

**Task#06**

Question statement

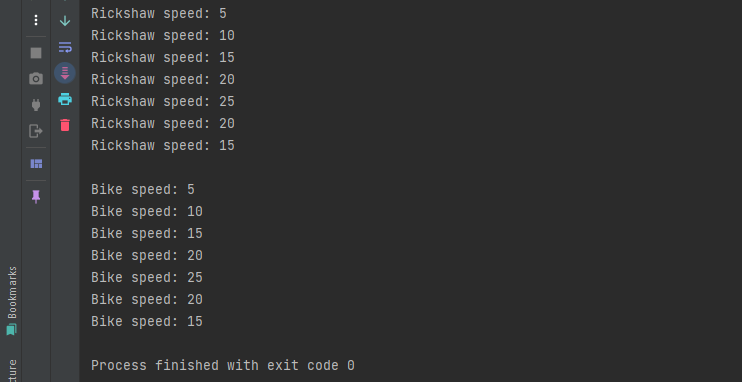
Write a class named Vehicle that can represent both the Rickshaw and Bike on the basis of number of wheels it has. Each vehicle has the following details • year. An int that holds the vehicle’s model year. • manufacturer. A string that holds the manufacturer name of that vehicle. • speed. An int that holds the vehicle’s current speed. In addition, the class should have the following member functions. • accelerate. The accelerate function should add 5 to the speed member variable each time itis called. • brake. The brake function should subtract 5 from the speed member variable each time it Is called. Demonstrate the class in a program that creates a Vehicle object for a Rickshaw and for a Bike both, and then calls the accelerate function five times. After each call to the accelerate function, get the current speed of the car and display it. Then, call the brake function two times. After each call to the brake function, get the current speed of the car and display it

# Q6.Java

**Code:**

class Vehicle{  
  
 int year, speed;  
 String manufacturer;  
  
 public void accelerate(){  
 this.speed += 5;  
 }  
  
 public void brake(){  
 this.speed -= 5;  
 }  
  
 public int getSpeed() {  
 return speed;  
 }  
  
} // end of class Vehicle  
  
public class Q6 {  
 public static void main(String[] args) {  
  
 Vehicle rickshaw = new Vehicle();  
 Vehicle bike = new Vehicle();  
  
 rickshaw.accelerate();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
 rickshaw.accelerate();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
 rickshaw.accelerate();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
 rickshaw.accelerate();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
 rickshaw.accelerate();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
  
 rickshaw.brake();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
 rickshaw.brake();  
 System.*out*.println("Rickshaw speed: "+rickshaw.getSpeed());  
  
 System.*out*.println();  
  
 bike.accelerate();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 bike.accelerate();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 bike.accelerate();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 bike.accelerate();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 bike.accelerate();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
  
 bike.brake();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 bike.brake();  
 System.*out*.println("Bike speed: "+bike.getSpeed());  
 } // end of main() method  
} // end of program

**Output:**

****