**Usman Institute of Technology**

**Department of Computer Science**

**Course Code: SE312**

**Course Title: Software Construction and Development**

**SPRING 2024**

**Lab 07**

**Objective:** Incorporating the code optimization techniques for perfect execution of the program.

**Student Information**

|  |  |
| --- | --- |
| Student Name | Syed Muhammad Zaid |
| Student ID | 20B-052-SE |
| Date | 5/10/2024 |

**Assessment**

|  |  |
| --- | --- |
| Marks Obtained |  |
| Remarks |  |
| Signature |  |

**LAB TASK:**

1. Create a design for the mark sheet by taking runtime value of student name, total marks,

obtained marks and calculate its percentage, grade and GPA. Use good practices of

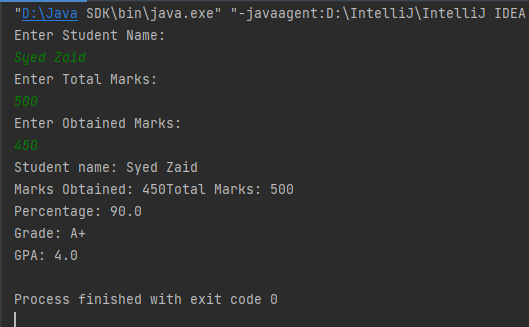
programming that we have studied and ensure that the outcomes should be presented in a

proper Viewable approach.

**CODE:**

/\*  
\*Create a design for the mark sheet by taking runtime value of student name, total marks,  
obtained marks and calculate its percentage, grade and GPA. Use good practices of  
programming that we have studied and ensure that the outcomes should be presented in a  
proper Viewable approach.  
\*/  
  
import java.util.Scanner;  
  
public class MarkSheet {  
  
 String student\_name;  
 int totalMarks;  
 int obtainedMarks;  
 double percentage;  
 String grade;  
 double GPA;  
 Scanner scanner = new Scanner(System.*in*);  
  
 public MarkSheet() {  
 // Class initialization  
 System.*out*.println("Enter Student Name: ");  
 this.student\_name = scanner.nextLine();  
 System.*out*.println("Enter Total Marks: ");  
 this.totalMarks = scanner.nextInt();  
 System.*out*.println("Enter Obtained Marks: ");  
 this.obtainedMarks = scanner.nextInt();  
 this.percentage = 0;  
 this.grade = null;  
 this.GPA = 0;  
 scanner.close();  
  
 }  
  
 public double calculatePercentage() {  
 this.percentage = ((double) this.obtainedMarks / this.totalMarks) \* 100;  
 return this.percentage;  
 }  
  
 public String calculateGrade() {  
 calculatePercentage();  
  
 if (this.percentage >= 90.0) {  
 this.grade = "A+";  
 } else if (this.percentage >= 80.0) {  
 this.grade = "A-";  
 } else if (this.percentage >= 70.0) {  
 this.grade = "A";  
 } else if (this.percentage >= 60.0) {  
 this.grade = "B";  
 } else if (this.percentage >= 50.0) {  
 this.grade = "C";  
 } else {  
 this.grade = "F";  
 }  
  
 return this.grade;  
 }  
  
 public double calculateGPA() {  
 if ("A+".equals(this.grade)) {  
 this.GPA = 4.0;  
 } else if ("A-".equals(this.grade)) {  
 this.GPA = 3.5;  
 } else if ("A".equals(this.grade)) {  
 this.GPA = 3.0;  
 } else if ("B".equals(this.grade)) {  
 this.GPA = 2.75;  
 } else if ("C".equals(this.grade)) {  
 this.GPA = 2.0;  
 } else {  
 this.GPA = 1.5;  
 }  
 return this.GPA;  
 }  
  
  
}

**OUTPUT:**

****

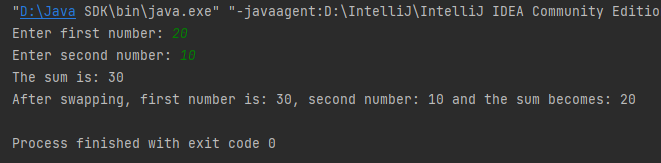
1. Create a class and take two input variables of integer type. Then perform addition of these two variables and store the result in new variable named sum.

After adding the values, swap the final summed value with one of the input values. Make sure to use proper naming conventions within your code.

**CODE:**

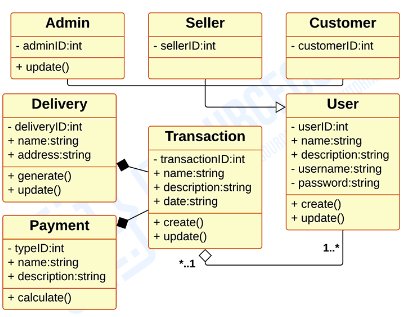
import java.util.Scanner;  
  
public class Swapping {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter first number: ");  
 int num1 = scanner.nextInt();  
  
 System.*out*.print("Enter second number: ");  
 int num2 = scanner.nextInt();  
  
 int sum = num1 + num2;  
 System.*out*.println("The sum is: " + sum);  
  
 int temp = num1;  
 num1 = sum;  
 sum = temp;  
  
 System.*out*.println("After swapping, first number is: "  
 + num1 + ", second number: " + num2 +  
 " and the sum becomes: " + sum);  
 scanner.close();  
 }  
}

**OUTPUT:**



**HOME TASKS**

1. Convert the following class diagrams into optimized and meaningful java code.



**CODE:**

public class Admin {  
 int adminID;  
 String adminName;  
  
 public Admin(int adminId, String adminName) {  
 this.adminID = adminId;  
 this.adminName = adminName;  
  
 }  
  
 public void update(){  
 System.*out*.println("The settings for admin are updated.");  
 }  
}

public class Seller {  
 int sellerID;  
 String sellerName;  
  
 public Seller(int sellerID, String sellerName){  
 this.sellerID = sellerID;  
 this.sellerName = sellerName;  
 }  
}

public class Customer {  
 int customerID;  
 String customerName;  
  
 public Customer(int customerID, String customerName) {  
 this.customerID = customerID;  
 this.customerName = customerName;  
 }  
}

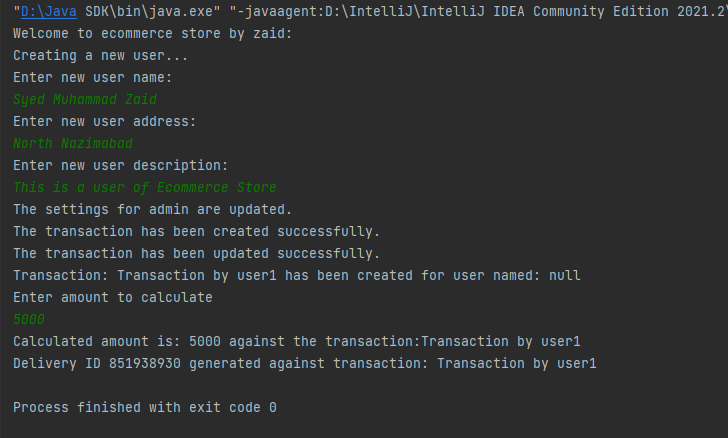
import java.util.UUID;  
import java.util.Scanner;  
  
public class Delivery {  
 int deliveryID;  
 String receiverName;  
 String receiverAddress;  
  
 public Delivery(int deliveryID, String receiverName, String receiverAddress) {  
 this.deliveryID = deliveryID;  
 this.receiverName = receiverName;  
 this.receiverAddress = receiverAddress;  
 }  
  
 public void generate(String transactionName) {  
 UUID delivery\_uuid = UUID.*randomUUID*();  
 this.deliveryID = Math.*abs*(delivery\_uuid.hashCode());  
 System.*out*.println("Delivery ID " + this.deliveryID + " generated against transaction: " + transactionName);  
 }  
  
  
 public void update(String newName) {  
 System.*out*.println("Enter new receiver name: ");  
 Scanner newReceiverName = new Scanner(System.*in*);  
 this.receiverName = newReceiverName.nextLine();  
 System.*out*.println("Enter new delivery address");  
 Scanner newAddress = new Scanner(System.*in*);  
 this.receiverAddress = newAddress.nextLine();  
 System.*out*.println("Customer details updated successfully.");  
 }  
  
}

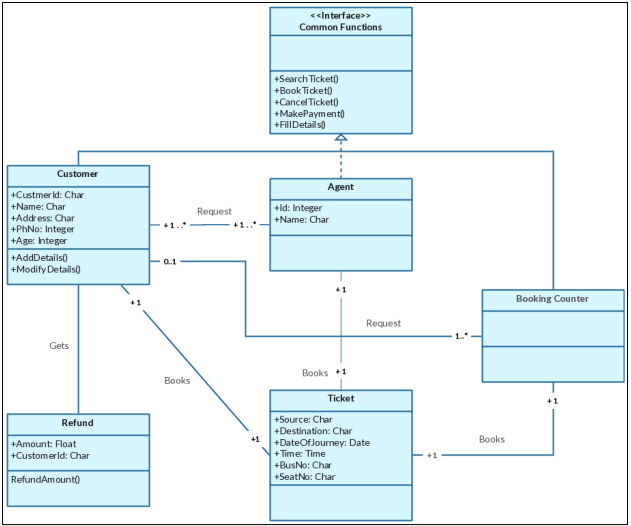
public class Transaction {  
 int transactionID;  
 String transactionName;  
 String transactionDesc;  
 String transactionDate;  
  
 public Transaction(int transactionID, String transactionName, String transactionDesc, String transactionDate) {  
 this.transactionID = transactionID;  
 this.transactionName = transactionName;  
 this.transactionDesc = transactionDesc;  
 this.transactionDate = transactionDate;  
 }  
  
 public void create() {  
 System.*out*.println("The transaction has been created successfully.");  
 }  
  
 public void update() {  
 System.*out*.println("The transaction has been updated successfully.");  
 }  
}

import java.util.Scanner;  
  
public class Payment {  
 int paymentID;  
 String senderName;  
 String senderDescription;  
  
 int amount;  
  
 public Payment(int paymentID, String senderName, String senderDescription) {  
 this.paymentID = paymentID;  
 this.senderName = senderName;  
 this.senderDescription = senderDescription;  
 this.amount = 0;  
 }  
  
 public void calculate(String transactionName) {  
 Scanner amount = new Scanner(System.*in*);  
 System.*out*.println("Enter amount to calculate");  
 this.amount = amount.nextInt();  
 System.*out*.println("Calculated amount is: " + this.amount + " against the transaction:" + transactionName);  
 amount.close();  
 }  
}

import java.util.Scanner;  
  
public class User {  
 private int userID;  
 String name;  
 String userAddress;  
 String userDescription;  
 private String userName;  
 private String userPassword;  
  
 public User() {  
 this.name = null;  
 this.userAddress = null;  
 this.userDescription = null;  
 this.userName = null;  
 this.userPassword = null;  
  
 }  
  
 public void createUser() {  
 Scanner inputUserDetails = new Scanner(System.*in*);  
 System.*out*.println("Creating a new user...");  
 System.*out*.println("Enter new user name: ");  
 this.name = inputUserDetails.nextLine();  
 System.*out*.println("Enter new user address: ");  
 this.userAddress = inputUserDetails.nextLine();  
 System.*out*.println("Enter new user description: ");  
 this.userDescription = inputUserDetails.nextLine();  
// inputUserDetails.close();  
 }  
  
 public void createTransaction(String transactionName) {  
 System.*out*.println("Transaction: " + transactionName + " has been created for user named: " + this.userName);  
 }  
  
 public void updateUser() {  
 Scanner inputUserDetails = new Scanner(System.*in*);  
 System.*out*.println("Updating an existing user...");  
 System.*out*.println("Enter new user name: ");  
 this.name = inputUserDetails.nextLine();  
 System.*out*.println("Enter new user address: ");  
 this.userAddress = inputUserDetails.nextLine();  
 System.*out*.println("Enter new user description: ");  
 this.userDescription = inputUserDetails.nextLine();  
 inputUserDetails.close();  
 }  
}

**OUTPUT:**

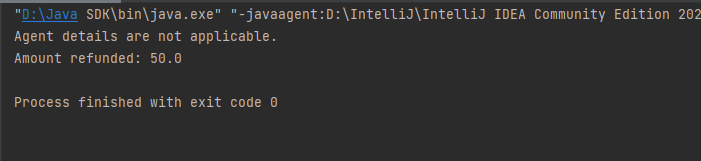
****



**CODE:**

import java.util.Date;  
import java.time.LocalTime;  
  
interface CommonFunctions {  
 void searchTicket();  
 void bookTicket();  
 void cancelTicket();  
 void makePayment();  
 void fillDetails();  
}  
  
class CustomerClass implements CommonFunctions {  
 char customerId;  
 char name;  
 char address;  
 int age;  
  
 @Override  
 public void searchTicket() { }  
  
 @Override  
 public void bookTicket() { }  
  
 @Override  
 public void cancelTicket() { }  
  
 @Override  
 public void makePayment() { }  
  
 @Override  
 public void fillDetails() { }  
  
 public void addDetails() { }  
  
 public void modifyDetails() { }  
}  
  
class Agent implements CommonFunctions {  
 int id;  
 char name;  
  
 @Override  
 public void searchTicket() { }  
  
 @Override  
 public void bookTicket() { }  
  
 @Override  
 public void cancelTicket() { }  
  
 @Override  
 public void makePayment() { }  
  
 @Override  
 public void fillDetails() {  
 System.*out*.println("Agent details are not applicable.");  
 }  
}  
  
class Ticket {  
 char source;  
 char destination;  
 Date dateOfJourney;  
 LocalTime time;  
 String busNo;  
 String seatNo;  
  
 public Ticket(char source, char destination, Date dateOfJourney, LocalTime time, String busNo, String seatNo) {  
 this.source = source;  
 this.destination = destination;  
 this.dateOfJourney = dateOfJourney;  
 this.time = time;  
 this.busNo = busNo;  
 this.seatNo = seatNo;  
 }  
}  
  
class Refund {  
 float amount;  
  
 public Refund(float amount) {  
 this.amount = amount;  
 }  
  
 public float refundAmount() {  
 return amount;  
 }  
}

**OUTPUT:**

****

**MAIN.JAVA IMPLEMENTATION:**

import java.util.Date;  
import java.time.LocalTime;  
  
class Ecommerce {  
 public static void main(String[] args) {  
 System.*out*.println("Welcome to ecommerce store by zaid: ");  
 // Handling User operations  
 User user1 = new User();  
 user1.createUser();  
  
 // Handling Admin operations  
 Admin admin = new Admin(001, "AdminOfEcommerceStore");  
 admin.update();  
  
 // Handling Seller operations  
 Seller seller = new Seller(1001, "Ali");  
  
 // Handling Customer operations  
 Customer customer1 = new Customer(2001, "Asad");  
  
 // Creating payment  
 Payment payment1 = new Payment(4001, "Asad", "This is payment by a customer.");  
  
 // Creating Delivery  
 Delivery delivery1 = new Delivery(9001, "Asad", "North Nazimabad");  
  
  
 // Creating & Updating Transaction for User,Delivery, Payment  
 Transaction transaction1 = new Transaction(5001, "Transaction by user1", "Transcation for user1", "5/10/24");  
 transaction1.create();  
 transaction1.update();  
  
 // Aggregating Transaction with User  
 user1.createTransaction(transaction1.transactionName);  
  
 // Composing payment with Transaction  
 payment1.calculate(transaction1.transactionName);  
  
 // Composing delivery with Transaction  
 delivery1.generate(transaction1.transactionName);  
  
  
 }  
}  
  
  
class BookingSystem {  
 public static void main(String[] args) {  
 // Creating instances of CustomerClass and Agent  
 CustomerClass customer = new CustomerClass();  
 Agent agent = new Agent();  
  
 // Calling methods of CustomerClass  
 customer.searchTicket();  
 customer.bookTicket();  
 customer.cancelTicket();  
 customer.makePayment();  
 customer.fillDetails();  
 customer.addDetails();  
 customer.modifyDetails();  
  
 // Calling methods of Agent  
 agent.searchTicket();  
 agent.bookTicket();  
 agent.cancelTicket();  
 agent.makePayment();  
 agent.fillDetails();  
  
 // Creating instance of Ticket  
 Date dateOfJourney = new Date();  
 LocalTime time = LocalTime.*now*();  
 Ticket ticket = new Ticket('A', 'B', dateOfJourney, time, "123", "1");  
  
 // Creating instance of Refund  
 Refund refund = new Refund(50.0f);  
  
 // Calling method of Refund  
 float refundedAmount = refund.refundAmount();  
 System.*out*.println("Amount refunded: " + refundedAmount);  
 }  
}  
  
class LabTask1 {  
 public static void main(String[] args) {  
  
 MarkSheet markSheet1 = new MarkSheet();  
 System.*out*.println("Student name: " + markSheet1.student\_name);  
 System.*out*.println("Marks Obtained: " + markSheet1.obtainedMarks + "Total Marks: " + markSheet1.totalMarks);  
 System.*out*.println("Percentage: " + markSheet1.calculatePercentage());  
 System.*out*.println("Grade: " + markSheet1.calculateGrade());  
 System.*out*.println("GPA: " + markSheet1.calculateGPA());  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 System.*out*.println("------------------------- LAB - TASKS -------------------------");  
 LabTask1.*main*(args);  
  
 System.*out*.println("------------------------- HOME - TASKS | Task - 1 -------------------------");  
 Ecommerce.*main*(args);  
  
 System.*out*.println("------------------------- HOME - TASKS | Task - 2 -------------------------");  
 BookingSystem.*main*(args);  
  
  
 }  
}