import java.util.\*;  
  
class enrolled //enrolled class to denote student details object  
{  
 String first\_name,last\_name,DOB,address,phone,mail,category,Id,course;  
 fees fee\_data;  
 enrolled(String first\_name,String last\_name,String DOB,String address,String phone,String mail,String category,String course,String Id) //enrolled constructor  
 {  
 this.first\_name=first\_name;  
 this.last\_name=last\_name;  
 this.DOB=DOB;  
 this.address=address;  
 this.phone=phone;  
 this.mail=mail;  
 this.category=category;  
 this.Id=Id;  
 this.course=course;  
 if(category=="Domestic") //calculates fees and calls fee constructor  
 fee\_data=new fees(3000,0);  
 else  
 fee\_data=new fees(6000,0);  
 }  
}  
  
class fees //fee class to denote object fee  
{  
 int fees;  
 int paid;  
 int remaining;  
 String transaction;  
 fees(int fees,int paid) //fee constructor  
 {  
 this.fees=fees;  
 this.paid=paid;  
 this.remaining=fees-paid;  
 }  
}  
  
public class student  
{  
 static Scanner *sc*=new Scanner(System.*in*); //Scanner object to take input  
 static HashMap<String,enrolled> *students*=new HashMap<>(); //Hash map to store student details  
 static int *student\_count*=0; //no.of students enrolled totally  
 static HashMap<String,String> *reports*=new HashMap<>(); //hashmap to store complaints  
 public static void main(String[]args)  
 {  
 int choice=-1; //choice to select from menu  
 while(choice!=8)  
 {  
 System.*out*.println("---------------------------------------"); //introduction and menu message  
 System.*out*.println(" Bright Future College Home ");  
 System.*out*.println("---------------------------------------");  
 System.*out*.println("1. Enroll Students\n2. Pay fees\n3. List students on a course\n4. Change course\n5. View student profile");  
 System.*out*.println("6. Remove student from course\n7. Management reporting\n8. Quit");  
 System.*out*.print("Enter your choice(0-8): ");  
 choice=*sc*.nextInt();  
 *sc*.nextLine();  
 switch(choice) //working based on choice selected, calls respective functions with needed inputs  
 {  
 case 1:*enroll*(); break;  
 case 2:String Id=*sinput*("enter student ID: ");*payFees*(Id); break;  
 case 3:int course=4; while(!(course<=3 && course>0)){System.*out*.println("enter valid Integer INPUT");course=*iinput*("Choose course: (1. Computing, 2. Accounting, 3. Business}: ");}*listCourse*(course); break;  
 case 4:String Id1=*sinput*("enter student ID: ");  
 int course1=4; while(!(course1<=3 && course1>0)){System.*out*.println("enter valid Integer INPUT");course1=*iinput*("Choose course: (1. Computing, 2. Accounting, 3. Business}: ");}*changeCourse*(Id1,course1);break;  
 case 5:String Id2=*sinput*("enter student ID: ");*viewStudentProfile*(Id2);break;  
 case 6:String Id3=*sinput*("enter student ID: ");*removeStudent*(Id3); break;  
 case 7:String Id4=*sinput*("enter student ID: ");*managementReporting*(Id4); break;  
 case 8:System.*out*.println("Exiting now..........");  
 }  
 }  
 }  
  
 static String sinput(String prompt) //method to take string input by displaying prompt  
 {  
 while(true) {  
 try { //to curb any errors  
 System.*out*.print(prompt);  
 return *sc*.nextLine();  
 }  
 catch (Exception e)  
 {  
 continue;  
 }  
 }  
 }  
  
 static Integer iinput(String prompt) //method to take integer input  
 {  
 boolean done=false;  
 int rinput=4;  
 System.*out*.println(prompt);  
 while(true) { //loop to suppress other data type inputs  
 if(*sc*.hasNextInt()) {  
 rinput = *sc*.nextInt();  
 break;  
 }  
 else  
 {  
 System.*out*.println("Enter valid integer! ");  
 }  
 }  
 *sc*.nextLine();  
 return rinput;  
 }  
  
 static void enroll()  
 {  
 System.*out*.print("Enter no.of enrollments: "); //no.of enrollments to loop  
 int enrolls=*sc*.nextInt();  
 for(int i=0;i<enrolls;i++)  
 {  
 System.*out*.print("Enter details student "+(i+1)+":\n"); //gets required details of students  
 String first\_name=*sinput*("Enter first name: ");  
 String last\_name=*sinput*("last name: ");  
 String DOB=*sinput*("Date of Birth(dd/mm/yyyy): ");  
 String address=*sinput*("Address(including postal code): ");  
 String phone=*sinput*("Phone number(country code and phone number without spaces): ");  
 String mail=*sinput*("E-mail: ");  
 int choice=4; while(!(choice<=3 && choice>0)){System.*out*.println("enter valid Integer INPUT");choice=*iinput*("student course: (1. Computing, 2. Accounting, 3. Business}: ");}  
 String course="",Id="";  
 if(choice==1){  
 course="Computing";  
 Id="COM";}  
 else if(choice==2){  
 course="Accounting";  
 Id="ACC";}  
 else{  
 course="Business";  
 Id="BUS";}  
 choice=3; while(!(choice<=2 && choice>0)){System.*out*.println("enter valid Integer INPUT");choice=*iinput*("Student category(1. Domestic or 2. International: ");};  
 String category="";  
 if(choice==1){  
 category="Domestic";  
 Id="D"+*student\_count*+Id; //generates ID based on student count, category and course  
 }  
 else{  
 category="International";  
 Id="I"+*student\_count*+Id;  
 }  
 System.*out*.println("\nStudent ID for "+first\_name+" is "+Id+"\n");  
 enrolled enrolling=new enrolled(first\_name,last\_name,DOB,address,phone,mail,category,course,Id); //calls enrolled constructor to create object for new student  
 *students*.put(Id,enrolling); //student details is mapped with ID and stored in HASHMAP  
 System.*out*.println("Pay at least "+enrolling.fee\_data.fees/3+" for registration");  
 while(enrolling.fee\_data.paid<enrolling.fee\_data.fees/3) //collects registration fees  
 {  
 *payFees*(Id);  
 }  
 *student\_count*++;  
 }  
 System.*out*.println(enrolls+" students enrolled");  
 }  
  
 static void payFees(String Id)  
 {  
 enrolled student=*students*.get(Id); //gets student details using Hashmap  
 if(student==null)  
 {  
 System.*out*.println("Check id and try again"); //If no student exist with ID prompts accordingly  
 return;  
 }  
 System.*out*.println("Fees remaining for "+student.first\_name+" is $"+student.fee\_data.remaining); //displays fees details  
 if(student.fee\_data.remaining==0)  
 System.*out*.println("Congrats, no dues!");  
 else{  
 int paying=*iinput*("Enter amount: "); //gets paying amount  
 student.fee\_data.paid+=paying;  
 student.fee\_data.remaining=student.fee\_data.remaining-paying; //computes new fee details   
 loop:while(true) { //runs loop to get valid integer input  
 int choice=3; while(!(choice<3 && choice>0)){System.*out*.println("enter valid Integer INPUT");choice=*iinput*("Select mode of payment to pay $"+paying+": 1. Bank transfer\t2.Paypal: ");}  
 switch (choice) {  
 case 1: //switch to run over correct payment method  
 System.*out*.println("Transfer amount to following XYZ bank account\nAccount no: xxxxxxxxxxxx\nIFSC code:XXXX000XXXX\nBeneficiary name: Bright Future College");  
 break loop;  
 case 2:  
 System.*out*.println("Transfer amount to following paypal Account\nE-mail:brightfuture@bfc.in\nBeneficiary name: Bright Future College");  
 break loop;  
 default:  
 System.*out*.println("Enter valid choice"); continue loop;  
 }  
 }  
 }  
 student.fee\_data.transaction=*sinput*("Enter transaction ID: "); //gets transaction ID to verify payment  
 }  
  
 static void listCourse(int choice)  
 {  
 String course; //identifies course based on choice to avoid typo errors  
 if(choice==1)  
 course="Computing";  
 else if(choice==2)  
 course="Accounting";  
 else  
 course="Business";  
 System.*out*.println("All students registered in "+course+" are below");  
 for(String k:*students*.keySet()) //runs loop over students hashmap and prints all students under course  
 {  
 if(*students*.get(k).course.equals(course))  
 System.*out*.println(*students*.get(k).first\_name+" "+*students*.get(k).last\_name+" "+*students*.get(k).mail);  
 }  
 }  
  
 static void changeCourse(String Id,int choice)  
 {  
 String course;  
 enrolled student=*students*.get(Id); //get students details from hashmap  
 if(student==null) //checks if user is valid  
 {  
 System.*out*.println("Check id and try again");  
 return;  
 }  
 if(choice==1) //gets course based on choice to avoid typo errors  
 course="Computing";  
 else if(choice==2)  
 course="Accounting";  
 else  
 course="Business";  
 String confirm=*sinput*("Do you want to change your course from "+student.course+" to "+course+"? Y/N: "); //prompts again for confirmation  
 if(confirm.equals("Y")||confirm.equals("y")){ //changes course on confirmation  
 System.*out*.println("changed your course successfully! congratulations");  
 student.course=course;}  
 else{ //cancels change on denying  
 System.*out*.println("Failed to change course try again");  
 }  
 }  
  
 static void viewStudentProfile(String Id)  
 {  
 enrolled student=*students*.get(Id); //gets details of student from hashmap  
 if(student==null) // verify student ID  
 {  
 System.*out*.println("Check id and try again");  
 return;  
 }  
 System.*out*.println("Student profile is printed below"); //displays student details  
 System.*out*.println("Course registered: "+student.course);  
 System.*out*.println("First Name: "+student.first\_name);  
 System.*out*.println("Last name: "+student.last\_name);  
 System.*out*.println("Date of Birth(dd/mm/yyyy): "+student.DOB);  
 System.*out*.println("Address: "+student.address);  
 System.*out*.println("Category: "+student.category);  
 System.*out*.println("Fees paid: "+student.fee\_data.paid);  
 }  
  
 static void removeStudent(String Id)  
 {  
 enrolled student=*students*.get(Id); //gets student details based on ID  
 if(student==null) //verifies student ID  
 {  
 System.*out*.println("Check id and try again");  
 return;  
 }  
 String confirm=*sinput*("Do you want to de-register the student? Y/N: "); //prompts for confirmation  
 if(confirm.equals("Y")||confirm.equals("y")){ //deregisters on confirmation  
 *students*.remove(Id);  
 System.*out*.println("student de-registered successfully");  
 }  
 else //cancels de-registering on denying  
 System.*out*.println("Failed to change course try again");  
  
 }  
  
 static void managementReporting(String Id)  
 {  
 *reports*.put(Id,*sinput*("Enter you grievance: ")); //gets complaint and stores it in reports hashmap  
 System.*out*.println("Your problem is escalated to management, it will be resolved soon. Sorry, for inconvenience");  
 }  
}