|  |  |
| --- | --- |
| Sr No | Program/ Output |
| 1 | /\*C++ program to create base class called STUDENT (name, reg no, age) and using  inheritance create derived classes UG student and PG student having field such as semester, fee  and stipend. Enter data of 5 students. Find average age semester wise all UG and PG students separately.\*/  //22CO16 Joshua Fernandes 13/11/2023  #include<iostream> #include <bits/stdc++.h>  using namespace std; class Student{  protected:  string name;  unsigned int age, reg\_no; static int student\_count;  public:  virtual void avg\_age()=0;  };  int Student::student\_count = 0;  class UG\_Student: protected Student{ int sem, fees, stipend;  static int avg; public:  void set\_UG(string nam, int ag, int se){ name=nam, age=ag, sem=se,  reg\_no=20231000;  }  void avg\_age(){avg+=age;} void  set\_regNO(){reg\_no=(++student\_count)+reg\_no;} void avg\_display(){cout<<avg/5;}  void calFEES(){ if(sem==1)  fees=40000, stipend=7000; else if(sem==2)  fees=45000, stipend=7500; else if(sem==3)  fees=50000, stipend=8000; else if(sem==4)  fees=55000, stipend=9000; else if(sem==5)  fees=57000, stipend=8000; else if(sem==6)  fees=59000, stipend=8000; else if(sem==7)  fees=60000, stipend=8500; else if(sem==8)  fees=50000, stipend=8500;  }  };  int UG\_Student::avg = 0; |

|  |  |
| --- | --- |
|  | class PG\_Student: protected Student{  int sem, fees, stipend; static int avg;  public:  void set\_UG(string nam, int ag, int se){ name=nam, age=ag, sem=se,  reg\_no=20235000;  }  void avg\_age(){avg+=age;} void  set\_regNO(){reg\_no=(++student\_count)+reg\_no;} void avg\_display(){cout<<avg/5;}  void calFEES(){ if(sem==1)  fees=40000, stipend=6000; else if(sem==2)  fees=45000, stipend=6500; else if(sem==3)  fees=50000, stipend=7000; else if(sem==4)  fees=55000, stipend=7500;  }  };  int PG\_Student::avg = 0; int main(){  string name, deg; int age, sem; UG\_Student ug[5]; PG\_Student pg[5];  cout<<"ENTER UG STUDENTS INFO,\n";  for(int i=0;i<5;i++){ cout<<"\nEnter name: "; getline(cin,name);  cout<<"Enter age : "; cin>>age;  do{  cout<<"Enter SEM (1-8): "; cin>>sem;  }while(sem>8);  ug[i].set\_UG(name, age, sem); ug[i].avg\_age();  ug[i].set\_regNO();  ug[i].calFEES();  cin.ignore();  }  system("cls");  cout<<"ENTER PG STUDENTS INFO,\n";  for(int i=0;i<5;i++){ cout<<"\nEnter name: "; getline(cin,name); |

|  |  |
| --- | --- |
|  | cout<<"Enter age : "; |
| cin>>age; |
| do{ |
| cout<<"Enter SEM (1-4): "; |
| cin>>sem; |
| }while(sem>4); |
| pg[i].set\_UG(name, age, sem); |
| pg[i].avg\_age(); |
| pg[i].set\_regNO(); |
| pg[i].calFEES(); |
| cin.ignore(); |
| } |
| cout<<"\nAverage age of UG students is : "; |
| ug[5].avg\_display(); |
| cout<<"\naverage age of PG students is : "; |
| pg[5].avg\_display(); |
| return 0; |
| } |
| Output: |
| ENTER UG STUDENTS INFO, |
| Enter name: ash tucker Enter age : 18  Enter SEM (1-8): 2 |
| Enter name: bruce willis Enter age : 18  Enter SEM (1-8): 2 |
| Enter name: natalie portman Enter age : 19  Enter SEM (1-8): 4 |
| Enter name: christian bale Enter age : 20  Enter SEM (1-8): 6 |
| Enter name: peter parker Enter age : 21  Enter SEM (1-8): 8 |
| ENTER PG STUDENTS INFO, |
| Enter name: Enter age : 23  Enter SEM (1-4): 2 |
| Enter name: josh dsa Enter age : 24  Enter SEM (1-4): 3 |
| Enter name: joel naik Enter age : 22  Enter SEM (1-4): 1 |
| Enter name: max park |

|  |  |
| --- | --- |
|  | Enter age : 21 Enter SEM (1-4): 1  Enter name: rick green Enter age : 24  Enter SEM (1-4): 4  Average age of UG students is : 19 average age of PG students is : 22 |
| 2 | /\*   1. Print 5.4321 in a 12 field width with trailing zeros and left justified 2. Print following using set f () functions –   xxxxx4.324e + 01   1. Print following using manipulators + ###1#####+1.0000   \*/  //22CO16 Joshua Fernandes 13/11/2023  #include<iostream> #include<iomanip>  using namespace std; int main(){  double a=5.4321;  cout.precision(10); cout<<setw(12)<<fixed<<a<<endl;  double b=43.24; cout.fill('x'); cout.precision(3);  cout.setf(ios::scientific,ios::floatfield); cout<<setw(14)<<b<<endl;  cout<<resetiosflags(ios::floatfield); double c = 1.0d;  cout.fill('#'); cout.setf(ios::showpos);  cout.setf(ios::internal,ios::adjustfield); cout<<setw(5)<<c; cout.setf(ios::showpoint); cout<<resetiosflags(ios::adjustfield); cout.precision(4); cout<<setw(12)<<fixed<<c<<endl;  return 0;  }  Output: |
| 5.4321000000  xxxxx4.324e+01  +###1#####+1.0000 |
| 3 | /\*C++ program using file data structure to create a database for railway reservation  system for „n‟ number of customers details: passenger-name, age, startplace, end-dest, regid, |

|  |  |
| --- | --- |
|  | coach-type(2A, 3A, SL). The program should provide the following options   1. Display the passenger‟s list for 2A, 3A & SL and store it as separate files 2. To display only a particular passenger record based on his regid.   \*/  //22CO16 Joshua Fernandes 13/11/2023  #include <iostream> #include <string> #include <iomanip> #include <fstream> #include <vector> #include <windows.h> #include <chrono> #include <thread> #include <algorithm>  using namespace std;  using namespace this\_thread; using namespace chrono;  class Passenger{  protected:  static int passengerCount;  string name, departure, destination, regid, coach\_type;  int age, storageStatus; public:  friend void displayList2A(vector<Passenger>);  friend void displayList3A(vector<Passenger>);  friend void displayListSL(vector<Passenger>);  friend void findByID(vector<Passenger>, string searchID);  friend void storeListsByCoach(vector<Passenger>&);  void newPassenger();  //Function to show passenger details; void showPassenger(){  cout<<"Name: "<<name<<endl; cout<<"Age:  "<<age<<setw(50)<<"Registration ID: "<<regid<<endl;  cout<<"Dep: "<<departure<<setw(40)<<"Dest: "<<destination<<endl;  cout<<"Coach Type: "<<coach\_type<<endl;  }  void storePassenger();  };  int Passenger :: passengerCount = 0; |

|  |  |
| --- | --- |
|  | //Function to set the details for a new passenger object  void Passenger :: newPassenger(){  system("cls"); storageStatus=0; cout<<setw(21)<<"NEW  PASSENGER\n";  regid = to\_string((++passengerCount)+202300000);  cout<<"Registration ID:  "<<regid<<endl;  cin.ignore(numeric\_limits<streamsize>::ma  x(), '\n');  cout<<"Passenger name: "; getline(cin, name);  //cin.ignore(numeric\_limits<streamsize>:: max(), '\n');  cout<<"Age: "; cin>>age; cin.ignore();  cout<<"Departure(place): "; getline(cin,departure); cout<<"Destination: "; getline(cin,destination);  do{  cout<<"Coach Type(2A,  3A, SL): ";  cin>>coach\_type;  }while(coach\_type!="2A" && coach\_type!="3A" && coach\_type!="SL");  }  //Function to display Coach Type 2A Passenger List void displayList2A(vector<Passenger> passengers){  system("cls"); for(unsigned int  i=0;i<passengers.size();i++){  if(passengers[i].coach\_type=="2A"){ passengers[i].showPassenger();  }  else  continue;  }  cout<<"[esc] close list"; do{  }while(\_getwch()!=27); return;  } |

|  |  |
| --- | --- |
|  | //Function to display Coach Tpye 3A Passegngers List  void displayList3A(vector<Passenger> passengers){ system("cls");  for(unsigned int i=0;i<passengers.size();i++){  if(passengers[i].coach\_type=="3A"){ passengers[i].showPassenger();  }  else  continue;  }  cout<<"[esc] close list"; do{  }while(\_getwch()!=27); return;  }  //Function to display Coach Tpye SL Passegngers List void displayListSL(vector<Passenger> passengers){  system("cls"); for(unsigned int  i=0;i<passengers.size();i++){  if(passengers[i].coach\_type=="SL"){ passengers[i].showPassenger();  }  else  continue;  }  cout<<"[esc] close list"; do{  }while(\_getwch()!=27); return;  }  //Function to store passenger lists in separate files based on the coach type  void storeListsByCoach(vector<Passenger> &passengers){  for(unsigned int i=0;i<passengers.size();i++){  if(passengers[i].coach\_type=="2A" && passengers[i].storageStatus==0){  passengers[i].storageStatus=1;  ofstream list2A("PassengerList2A.txt", ios::app); |

|  |  |
| --- | --- |
|  | list2A << "Name: " << passengers[i].name << "\n"  <<  "Age: " << passengers[i].age << "\t\t\tRegistration ID: " << passengers[i].regid << "\n"  <<  "Dep: " << passengers[i].departure << "\t\tDest: " << passengers[i].destination << "\n"  <<  "Coach Type: " << passengers[i].coach\_type << "\n\n";  list2A.close();  }  else if(passengers[i].coach\_type=="3A" && passengers[i].storageStatus==0){  passengers[i].storageStatus=1;  ofstream list3A("PassengerList3A.txt", ios::app);  list3A << "Name: " << passengers[i].name << "\n"  << "Age: " << passengers[i].age << "\t\t\tRegistration ID: " << passengers[i].regid << "\n"  << "Dep: " << passengers[i].departure << "\t\tDest: " << passengers[i].destination << "\n"  << "Coach Type: " << passengers[i].coach\_type << "\n\n";  list3A.close();  }  else if(passengers[i].coach\_type=="SL" && passengers[i].storageStatus==0){  passengers[i].storageStatus=1;  ofstream listSL("PassengerListSL.txt", ios::app);  listSL << "Name: " << passengers[i].name << "\n"  <<  "Age: " << passengers[i].age << "\t\t\tRegistration ID: " << passengers[i].regid << "\n"  <<  "Dep: " << passengers[i].departure << "\t\tDest: " << passengers[i].destination << "\n"  <<  "Coach Type: " << passengers[i].coach\_type << "\n\n";  listSL.close();  } |

|  |  |
| --- | --- |
|  | else  continue;  }  cout<<"Passenger details saved to file."; cout<<"[esc] close";  do{  }while(\_getwch()!=27); return;  }  //Function to find passenger by id  void findByID(vector<Passenger> passengers, string searchID){  for(unsigned int i=0; i<passengers.size();i++){  if(passengers[i].regid==searchID){  passengers[i].showPassenger();  cout<<"[esc] close"; do{  }while(\_getwch()!=27); return;  }  }  cout<<"Passenger ID not found.\n(Make sure the ID you have entered is a valid ID.)";  sleep\_for(milliseconds(3000)); return;  }  void readListFile(string fileName) { fileName+=".txt";  ifstream file(fileName.c\_str());  if (!file) {  cerr << "Error opening file: " << fileName << endl;  return;  }  string line;  while (getline(file, line)) { cout << line << endl;  }  file.close();  cout << "[esc] close"; do {  } while (\_getwch() != 27);  return;  }  int main(){  int c, n; |

|  |  |
| --- | --- |
|  | cout<<"Enter number of passengers: "; cin>>n;  vector <Passenger> passengers(n); for(int i=0;i<n;i++){  passengers[i].newPassenger();  }  do{  system("cls");  cout<<"1. Display 2A Passengers List\n2. Display 3A Passenger List\n3. Display SL Passenger List";  cout<<"\n4. Store passenger Lists by Coach Type\n5. Find passenger by Registration ID";  cout<<"\n6. Add new passenger\n7. Read passenger lists from file\n8. Exit\nEnter choice: ";  cin>>c; switch(c){  case 1: displayList2A(passengers);break;  case 2: displayList3A(passengers);break;  case 3: displayListSL(passengers);break;  case 4: storeListsByCoach(passengers);break;  case 5: {string searchID;  cout<<"Enter registration ID to  find: ";  cin>>searchID;  findByID(passengers, searchID); break;}  case 6: {Passenger  passenger;  passenger.newPassenger(); passengers.push\_back(passenger);  break;}  case 7: {  int x;  cout<<"\n\n1. 2A Coach passenger List.\n2. 3A Coach passenger List.\n3. SL Coach Passenger List\nChoose list: ";  cin>>x; switch(x){  case 1: readListFile("PassengerList2A");break; |

|  |  |
| --- | --- |
|  | case 2: readListFile("PassengerList3A");break;  case 3: readListFile("PassengerListSL");break;  }  }  }  }while(c!=8);  return 0;  }  Output: |
| Enter number of passengers: 3  NEW PASSENGER  Registration ID: 202300001 Passenger name: Ryan Reynolds Age: 45  Departure(place): Goa Destination: Mumbai  Coach Type(2A, 3A, SL): 2A  NEW PASSENGER  Registration ID: 202300002 Passenger name: Jackson  Age: 20  Departure(place): Goa Destination: Delhi  Coach Type(2A, 3A, SL): 3A  NEW PASSENGER  Registration ID: 202300003 Passenger name: Matt Damon Age: 29  Departure(place): Goa Destination: Pakistan  Coach Type(2A, 3A, SL): SL   1. Display 2A Passengers List 2. Display 3A Passenger List 3. Display SL Passenger List 4. Store Passenger Lists by Coach Type 5. Find passenger by Registration ID 6. Add new Passenger 7. Read passenger lists from file 8. Exit   Enter choice: 1  Name: Ryan Reynolds Age: 45  Registration ID: 202300001 Dep: Goa  Dest: Mumbai Coach Type: 2A  [esc] close list |

|  |  |
| --- | --- |
|  | Name: Jackson  Age: 20  Registration ID: 202300002 Dep: Goa  Dest: Delhi Coach Type: 3A  [esc] close list  Name: Matt Damon Age: 29  Registration ID: 202300003 Dep: Goa  Dest: Pakistan  Coach Type: SL  [esc] close list   1. Display 2A Passengers List 2. Display 3A Passenger List 3. Display SL Passenger List 4. Store Passenger Lists by Coach Type 5. Find passenger by Registration ID 6. Add new Passenger 7. Read passenger listsfrom file 8. Exit   Enter choice: 4  Passenger details saved to file.[esc] close   1. 2A Coach Passenger List. 2. 3A Coach Passenger List. 3. SL Coach Passenger List 1   Name: Ryan Reynolds  Age: 45 Registration ID: 202300001  Dep: Goa Dest: Mumbai Coach Type: 2A  [esc] close   1. 2A Coach Passenger List. 2. 3A Coach Passenger List. 3. SL Coach Passenger List 2   Name: Jackson  Age: 20 Registration ID: 202300002  Dep: Goa Dest: Delhi Coach Type: 3A  [esc] close   1. 2A Coach Passenger List. 2. 3A Coach Passenger List. 3. SL Coach Passenger List 3   Name: Matt Damon  Age: 29 Registration ID: 202300003  Dep: Goa Dest: Pakistan  Coach Type: SL [esc] close |

|  |  |
| --- | --- |
|  | 1. Display 2A Passengers List 2. Display 3A Passenger List 3. Display SL Passenger List 4. Store Passenger Lists by Coach Type 5. Find passenger by Registration ID 6. Add new passenger 7. Read passenger lists from file 8. Exit   Enter choice:5  Enter registration ID to find: 202300006  Passenger ID not found.  (Make sure the ID you have entered is a valid ID.)   1. Display 2A Passengers List 2. Display 3A Passenger List 3. Display SL Passenger List 4. Store Passenger Lists by Coach Type 5. Find passenger by Registration ID 6. Add new passenger 7. Read passenger lists from file 8. Exit   Enter choice: 5  Enter registration ID to find: 202300002  Name: Jackson  Age: 20  Registration ID: 202300002 Dep: Goa  Dest: Delhi Coach Type: 3A [esc] close   1. Display 2A Passengers List 2. Display 3A Passengers List 3. Display SL Passengers List 4. Store Passenger Lists by Coach Type 5. Find passenger by Registration ID 6. Add new passenger 7. Read passenger lists from file 8. Exit   Enter choice: 8  Process returned 0 (0x0) execution time : 961.773 s  Press any key to continue. |