**ÔN TẬP**

**Môn học: Lập trình hướng đối tượng và C++**

|  |  |
| --- | --- |
| STT | Nội dung câu hỏi |
|  | a) Cài đặt lớp Vector gồm n thành phần là các số thực với các phương thức sau:  + Hàm tạo mặc định;  + Hàm hủy;  + Hàm nhập;  + Hàm xuất;  + Hàm toán tử + hai Vector;  + Hàm toán tử \* hai Vector;  + Hàm toán tử chỉ số.  b) Viết hàm main sử dụng lớp Vector đã xây dựng để nhập vào một mảng các Vector có cùng số chiều sau đó:  + In ra danh sách các Vector vừa nhập;  + Tính và in ra tổng của các Vector. |
|  | Xây dựng lớp các hàng hóa gồm các thuộc tính mã hàng hóa (số nguyên), tên hàng hóa, giá nhập, giá bán, số lượng và các phương thức:  + Hàm tạo mặc định;  + Hàm hủy;  + Hàm nhập;  + Hàm xuất;  + Hàm tính lợi nhuận = (giá bán – giá mua) \* số lượng bán.  b) Viết hàm main sử dụng lớp hàng hóa đã xây dựng nhập vào một mảng các hàng hóa sau đó:  + In ra danh sách các hàng hóa vừa nhập;  + Tính và in ra tổng lợi nhuận của các hàng hóa;  + In ra danh sách tối đa 10% mặt hàng bán lãi nhất có số lượng bán > 5.  CODE  #include <iostream>  using namespace std;  class HangHoa  {  protected:  int MaHang;  char TenHang[50];  int Gianhap;  int Giaban;  int Soluong;  public:  HangHoa(){}  void Nhap()  {    cout<<"Nhap ma hang : ";cin>>MaHang;  fflush(stdin);  cout<<"Gia nhap : ";cin>>Gianhap;  cout<<"Gia ban : ";cin>>Giaban;  cout<<"So luong : "; cin>>Soluong;  }  void Xuat()  {  cout<<"Ma hang : "<<MaHang<<endl;  cout<<"Ten Hang : "<<TenHang<<endl;  cout<<"Gia nhap : "<<Gianhap<<endl;  cout<<"Gia ban : "<<Giaban<<endl;  cout<<"So luong : "<<Soluong<<endl;  }  ~ HangHoa(){}  int LoiNhuan()  {  return (Giaban - Gianhap)\*Soluong;  }  int Sl()  {  return Soluong;  }  };  int main()  {  int n ;  cout<<"Nhap so luong hang hoa : "<<endl;  cin>>n;  HangHoa \*a = new HangHoa[n+1];  // n+1 vi co ham khoi tao ban dau  for (int i=1;i<=n;i++)  {  a[i].Nhap();  }  for (int i=1;i<=n;i++)  {  a[i].Xuat();  cout<<"Loi nhuan : "<<a[i].LoiNhuan();  }  for (int i=1;i<=n;i++)  {  for (int j=2;j<=n;j++)  if(a[i].LoiNhuan()<a[j].LoiNhuan())  {  HangHoa temp = a[i];  a[i] = a[j];  a[j] = temp;  }  }  for (int i=1;i<=n;i++)  {  a[i].Xuat();  cout<<"Loi nhuan : "<<a[i].LoiNhuan();  }  int i = 1;  while ((i/n)<=0.1)  {  if(a[i].Sl()>5)  {  cout<<"Loi nhuan : "<<a[i].LoiNhuan();    }  i++;  }    } |
|  | Một cửa hàng bán thuốc lưu thông tin các loại thuốc gồm: Tên thuốc, mã thuốc, số lượng và giá bán. Theo quan điểm Lập trình HĐT bạn hãy viết chương trình thực hiện:  a) Nhập và in ra màn hình danh sách các loại thuốc có trong cửa hàng;  b) Sắp xếp danh sách các loại thuốc theo tên, cùng tên theo mã.  #include <iostream>  #include <string.h>  using namespace std;  class Thuoc  {  protected:  char MaThuoc[10];  char TenThuoc[50];  int Giaban;  int Soluong;  public:  HangHoa(){}  void Nhap()  {  fflush(stdin);  cout<<"Nhap ma thuoc : ";cin.getline(MaThuoc,10);  cout<<"Ten thuoc : ";cin.getline(TenThuoc,50);  cout<<"Gia ban : ";cin>>Giaban;  cout<<"So luong : "; cin>>Soluong;  }  void Xuat()  {  cout<<"Ma thuoc : "<<MaThuoc<<endl;  cout<<"Ten thuoc : "<<TenThuoc<<endl;  cout<<"Gia ban : "<<Giaban<<endl;  cout<<"So luong : "<<Soluong<<endl;  }  char \*getTenThuoc()  {  return TenThuoc;  }  char \*getMaThuoc()  {  return MaThuoc;  }  };  int main()  {  int n ;  cout<<"Nhap so luong thuoc : "<<endl;  cin>>n;  Thuoc \*a = new Thuoc[n+1];  // n+1 vi co ham khoi tao ban dau  for (int i=1;i<=n;i++)    a[i].Nhap();  }  for (int i=1;i<=n;i++)  {  a[i].Xuat();  }  for (int i=1;i<=n;i++)  {  for (int j=2;j<=n;j++)  if((strcmp(a[i].getTenThuoc(),a[j].getTenThuoc())>0) ||  (strcmp(a[i].getTenThuoc(),a[j].getTenThuoc())==0)  && (strcmp(a[i].getMaThuoc(),a[j].getMaThuoc())>0))  {  Thuoc temp = a[i];  a[i] = a[j];  a[j] = temp;  }  }  cout<<"\*\*\*\*\*\*\*\*\*\*\*\* : "<<endl;  for (int i=1;i<=n;i++)  {  a[i].Xuat();  }  } |
|  | Thông tin của nhân viên trong một công ty gồm: Họ tên, Năm sinh, Lương cơ bản (lương một ngày công), Số ngày công.  a) Anh (chị) hãy xây dựng lớp Nhan\_Vien của công ty với các thuộc tính trên. Lớp Nhan\_Vien có các phương thức: Khởi tạo, Nhập thông tin, In thông tin, Tính lương tháng của nhân viên (theo công thức: lương tháng = số ngày công \* lương cơ bản).  b) Xây dựng lớp Bang\_Luong với các thuộc tính: Số lượng nhân viên, Danh sách nhân viên. Lớp Bang\_Luong có các phương thức: Khởi tạo (Khởi tạo một danh sách rỗng), Hủy bỏ (hủy danh sách), Nhập danh sách, In danh sách, Thống kế các nhân viên có lương tháng < 5.000.000.  #include <iostream>  using namespace std;  class CongTy  {  protected:  char HoTen[50];  int NamSinh;  int LuongCb;  int SoNgayCong;  public:  CongTy(){}  void Nhap()  {  fflush(stdin);  cout<<"Ho ten : ";cin.getline(HoTen,50);  cout<<"Nam sinh : ";cin>>NamSinh;  cout<<"Luong co ban : "; cin>>LuongCb;  cout<<"So ngay cong : "; cin>>SoNgayCong;  }  void Xuat()  {  cout<<"Ho ten : "<<HoTen<<endl;  cout<<"Nam sinh : "<<NamSinh;  cout<<"Luong co ban : "<<LuongCb;  cout<<"So ngay cong : "<<SoNgayCong;  cout<<"Luong : "<<LuongNV();  }  int LuongNV()  {  return SoNgayCong\*LuongCb;  }  };  class Bang\_Luong  {  protected:  int SoLuongNV;  public:  Bang\_Luong(){}  void Nhap()  {  cout<<"Nhap so luong nhan vien :";cin>>SoLuongNV;  CongTy \*nv = new CongTy[SoLuongNV+1];  //Them 1 gia tri o hom khoi tao  for (int i=1;i<=SoLuongNV;i++)  {  nv[i].Nhap();  }  }  void Xuat()  {  CongTy \*nv;  for (int i=1;i<=SoLuongNV;i++)  {  nv[i].Xuat();  }  }  void InLuongduoi5tr()  {  CongTy \*nv;  for (int i=1;i<=SoLuongNV;i++)  {  if(nv[i].LuongNV()<5000000)  {  nv[i].Xuat();  }  }  }  ~Bang\_Luong(){};  };  int main()  {  Bang\_Luong nv;  nv.Nhap();  nv.Xuat();  nv.InLuongduoi5tr();  } |
|  | Cho thông tin của một sinh viên gồm: Tên sinh viên, mã sinh viên, điểm trung bình. Bạn hãy xây dựng các lớp cần thiết để có thể:  a) Nhập, in danh sách gồm n sinh viên.  b) Tìm sinh viên có điểm trung bình lớn nhất. Sắp xếp sinh viên theo tên, cùng tên theo mã sinh viên.  CODE :  #include <iostream>  #include <string.h>  using namespace std;  class SinhVien  {  protected:  char MaSV[10];  char TenSV[50];  float DiemTB;  public:  CongTy(){}  void Nhap()  {  fflush(stdin);  cout<<"Ma SV : ";cin.getline(MaSV,10);  cout<<"Ten SV : ";cin.getline(TenSV,50);  cout<<"Diem trung binh : ";cin>>DiemTB;  }  void Xuat()  {  fflush(stdin);  cout<<"Ma SV : "<<MaSV;  cout<<"Ten SV : "<<TenSV;  cout<<"Diem trung binh : "<<DiemTB;  }  float getDiemTB()  {  return DiemTB;  }  char \*getTenSV()  {  return TenSV;  }  char \*getMaSV()  {  return MaSV;  }  };  int main()  {  int n;  cout<<"Nhap so luong sinh vien :";cin>>n;  SinhVien \*sv = new SinhVien[n+1];  //Them 1 gia tri o hom khoi tao  for (int i=1;i<=n;i++)  {  sv[i].Nhap();  }  for (int i=1;i<=n;i++)  {  sv[i].Xuat();  }  //Max  float max =0;  for (int i=1;i<=n;i++)  {  if(sv[i].getDiemTB()<sv[i+1].getDiemTB())  {  max = sv[i+1].getDiemTB();  }  }  //SX  for (int i=1;i<=n;i++)  {  for (int j=2;j<=n;j++)  if((strcmp(sv[i].getTenSV(),sv[j].getTenSV())>0) ||  (strcmp(sv[i].getTenSV(),sv[j].getTenSV())==0)  && (strcmp(sv[i].getMaSV(),sv[j].getMaSV())>0))  {  SinhVien temp = sv[i];  sv[i] = sv[j];  sv[j] = temp;  }  }  } |
|  | Xây dựng lớp các ma trận các số nguyên có kích thước 2x2 với các phương thức:  + Hàm nhập;  + Hàm xuất;  + Hàm tính định thức;  + Hàm cộng hai ma trận;  + Hàm toán tử chỉ số.  Viết hàm main nhập vào ma trận sau đó:  + In ra các ma trận vừa nhập;  + In ra tổng của các ma trận. |
|  | Xây dựng lớp các hình tròn gồm các thuộc tính dữ liệu là tọa độ tâm và bán kính với các phương thức:  + Hàm khởi tạo mặc định;  + Hàm nhập:  + Hàm xuất;  + Hàm tính diện tích;  + Hàm kiểm tra một điểm có nằm trong đường tròn hay không  Nhập vào một mảng các hình tròn và 1 điểm sau đó:  + In ra mảng các hình tròn vừa nhập;  + Sắp xếp các hình tròn theo chiều giảm dần của diện tích;  + In ra tối đa 10% các hình tròn có diện tích lớn nhất và không chứa điểm đã nhập.  CODE:  #include<iostream>  #include<cmath>  using namespace std;  class HinhTron  {  int x,y,r;  public:  HinhTron(){}  void Nhap()  {  cout<<"Nhap toa do tam (x,y) "<<endl;  cout<<"x : ";cin>>x;cout<<endl;  cout<<"y : ";cin>>y;cout<<endl;  cout<<"Nhap ban kinh : ";cin>>r;cout<<endl;  }  void Xuat()  {  cout<<endl<<"Toa Do(x,y) = "<< "("<<x<<","<<y<<")"<<endl;  cout<<"Ban kinh = " <<r<< endl;  cout<<"DienTich = "<<DienTich();  }  float DienTich()  {  return (float)(3.14159\*r\*r);  }  bool KT(int a,int b)  {  float d1;  d1 = sqrt((a-x)\*(a-x)+(b-y)\*(b-y));  if (r==d1)  {  return true; // A nam tren duong tron  }  if(r>d1)  {  return true; // A nam trong duong tron  }  if(r<d1)  {  return false; // A nam ngoai duong tron  }  }  };  int main()  {  int n;  cout<<"Nhap so luong hinh tron :";cin>>n;cout<<endl;  HinhTron \*ht = new HinhTron[n];  cout<<"\*\*Nhap hinh tron\*\*"<<endl;  for (int i = 1 ; i<n ; i++)  {  ht[i].Nhap();  }  cout<<"\*\*Nhap toa do A(x1,y1)\*\*"<<endl;  int x1,y1;  cout<<"x1 = ";cin>>x1;cout<<endl;  cout<<"y1 = ";cin>>y1;cout<<endl;  cout<<"\*\*Xuat hinh tron\*\*"<<endl;  for (int i = 0 ; i<n ; i++)  {  ht[i].Xuat();  }  cout<<endl<<"\*\*Sap xep hinh tron\*\*"<<endl;  for (int i = 1 ; i<=n ; i++)  {  for (int j = 2 ; j<=n;j++)  if(ht[i].DienTich()<ht[j].DienTich())  {  HinhTron temp = ht[i];  ht[i] = ht[j];  ht[j] = temp;  }  }  cout<<"\*\*Xuat hinh tron\*\*"<<endl;  for (int i = 0 ; i<n ; i++)  {  ht[i].Xuat();  }  cout<<"\*\*In ra toi da 10% cac hinh tron\*\*"<<endl;  int i = 1;  do{  if(ht[i].KT(x1,y1)==false) // In ra hinh tron khong chua diem A  {  ht[i].Xuat();  }  i++;    }while((i/n)<=0.1);  return 0;  } |
|  | Cài đặt lớp học viên gồm các thuộc tính dữ liệu mã học viên, họ tên, ngành học và các phương thức:  + Hàm tạo mặc định và hàm tạo có tham số;  + Hàm nhập;  + Hàm xuất.  Hãy nhập vào một mảng n học viên và sau đó:  + In ra danh sách học viên vừa nhập;  + Nhập vào một ngành học và in ra danh sách các học viên học theo ngành đó;  + Sắp xếp danh sách học viên tăng dần theo họ tên, cùng họ tên theo ngành học và in kết quả ra màn hình  CODE :  #include <iostream>  #include <string.h>  using namespace std;  class HocVien  {  protected:  char MaHV[10];  char TenHV[50];  char NganhHoc[100];  public:  SinhVien(){}  void Nhap()  {  fflush(stdin);  cout<<"Ma SV : ";cin.getline(MaHV,10);  cout<<"Ten SV : ";cin.getline(TenHV,50);  cout<<"Nganh hoc : ";cin.getline(NganhHoc,100);  }  void Xuat()  {  fflush(stdin);  cout<<"Ma SV : "<<MaHV;  cout<<"Ten SV : "<<TenHV,50;  cout<<"Nganh hoc : "<<NganhHoc;  }  char \*getTenHV()  {  return TenHV;  }  char \*getNganhHoc()  {  return NganhHoc;  }  };  int main()  {  int n;  cout<<"Nhap so luong hoc vien :";cin>>n;  HocVien \*hv = new HocVien[n+1];  //Them 1 gia tri o hom khoi tao  for (int i=1;i<=n;i++)  {  hv[i].Nhap();  }  for (int i=1;i<=n;i++)  {  hv[i].Xuat();  }  //Nganh hoc  char nh[100];  cout<<"Nhap nganh hoc de xem : ";  cout<<"Nganh hoc : ";cin.getline(nh,100);    for (int i=1;i<=n;i++)  {  if((strcmp(hv[i].getTenHV(),nh)==0))  {  hv[i].Xuat();  }  }    //SX  for (int i=1;i<=n;i++)  {  for (int j=2;j<=n;j++)  if((strcmp(hv[i].getTenHV(),hv[j].getTenHV())>0) ||  (strcmp(hv[i].getTenHV(),hv[j].getTenHV())==0)  && (strcmp(hv[i].getNganhHoc(),hv[j].getNganhHoc())>0))  {  HocVien temp = hv[i];  hv[i] = hv[j];  hv[j] = temp;  }  }  for (int i=1;i<=n;i++)  {  hv[i].Xuat();  }  } |
|  | a) Xây dựng lớp các nhân viên gồm các thuộc tính dữ liệu họ tên, hệ số lương, phụ cấp với các phương thức nhập, xuất và tính lương (lương = hệ số lương \* 1310 + phụ cấp). Kế thừa lớp nhân viên xây dựng lớp Cán bộ có thêm 1 thuộc tính hệ số chức vụ với các phương thức như lớp nhân viên biết lương cán bộ = (hệ số lương + hệ số chức vụ) \* 1310 + phụ cấp.  b) Viết hàm main nhập vào một mảng n Nhân viên và Cán bộ sau đó:  + In ra danh sách vừa nhập;  + In ra danh sách tối đa 10% nhân viên có thu nhập thấp nhất và có hệ số lương nhỏ hơn 3.0 để nhận trợ cấp khó khăn cuối năm.  CODE:  #include<iostream>  using namespace std;  class NhanVien  {  protected:  char Hoten[50];  float Hesoluong;  float Phucap;  public :  NhanVien(){}  void Nhap()  {  cout<<"Nhap ho ten : ";cin.getline(Hoten,50);  cout<<"He so luong : ";cin>>Hesoluong;  cout<<"Phu cap : ";cin>>Phucap;  }  void Xuat()  {  cout<<"Ho ten : "<<Hoten;  cout<<"He so luong : "<<Hesoluong;  cout<<"Phu cap : "<<Phucap;  }  float LuongNhanVien()  {  return Hesoluong\*1310 + Phucap;  }  float getHesoluong()  {  return Hesoluong;  }  float getPhucap()  {  return Phucap;  }  };  class CanBo : public NhanVien  {  protected:  float HesoChucVu;  public :  CanBo(){}  void Nhap()  {  NhanVien::Nhap();  cout<<"He so chuc vu : ";cin>>HesoChucVu;  }  void Xuat()  {  NhanVien::Xuat();  cout<<"He so chuc vu : "<<HesoChucVu;  }  float LuongCanBo()  {  return (NhanVien::getHesoluong() + HesoChucVu)\*1310 + NhanVien::getPhucap();  }  };  int main()  {  int n,m;  //  cout<<"Nhap so luong nhan vien : ";cin>>n;  NhanVien \*nv= new NhanVien[n+1];  for (int i = 1;i<=n;i++)  {  nv[i].Nhap();  }  for (int i = 1;i<=n;i++)  {  nv[i].Xuat();  }  cout<<"Nhap so luong can bo : ";cin>>m;  CanBo \*cb = new CanBo[m+1];  for (int i = 1;i<=m;i++)  {  cb[i].Nhap();  }  for (int i = 1;i<=m;i++)  {  cb[i].Xuat();  }  //  int i =0;  while((i/n)<=0.1)  {  if(nv[i].getHesoluong()>3.0)  {  nv[i].Xuat();  }  i++;  }  return 0;  } |
|  | a) Xây dựng lớp các thí sinh xét tuyển đại học gồm các thành viên dữ liệu họ tên, điểm trung bình, điểm tiếng Anh với các phương thức nhập, xuất dữ liệu, xét tuyển (trả về true nếu thí sinh trúng tuyển: điểm trung bình >= 7.0 và điểm tiếng Anh >= 400). Kế thừa từ lớp thí sinh để xây dựng lớp thí sinh ưu tiên có thêm 1 trường dữ liệu là điểm ưu tiên với các phương thức như lớp thí sinh và tiêu chí trúng tuyển là điểm trung bình + điểm ưu tiên >= 7.0 và điểm tiếng Anh >= 400.  b) Viết hàm main nhập vào một mảng các thí sinh (bình thường và ưu tiên) đã xây dựng ở câu a sau đó:  + In ra danh sách các thí sinh vừa nhập;  + In ra danh sách các thí sinh trúng tuyển không vượt quá 30% tổng số thí sinh.  CODE :  #include<iostream>  using namespace std;  class XetTuyen  {  protected:  char Hoten[50];  float DiemTrungBinh;  float DiemTiengAnh;  bool xt;  public :  XetTuyen(){}  void Nhap()  {  fflush(stdin);  cout<<"Nhap ho ten : ";cin.getline(Hoten,50);  cout<<"Diem trung binh : ";cin>>DiemTrungBinh;  cout<<"Diem tieng anh : ";cin>>DiemTiengAnh;  if(DiemTrungBinh >=7.0 && DiemTiengAnh >= 400)  {  xt = true;  }  else {  xt = false;  }  }  void Xuat()  {  cout<<"Nhap ho ten : "<<Hoten;  cout<<"Diem trung binh : "<<DiemTrungBinh;  cout<<"Diem tieng anh : "<<DiemTiengAnh;  }  float getDiemTrungBinh()  {  return DiemTrungBinh;  }  bool getxt()  {  return xt;  }  };  class UuTien : public XetTuyen  {  protected:  float DiemUuTien;  public :  UuTien(){}  void Nhap()  {  XetTuyen::Nhap();  cout<<"Diem uu tien : ";cin>>DiemUuTien;  if(DiemTT() >=7.0 && DiemTiengAnh >= 400)  {  xt = true;  }  else {  xt = false;  }  }  void Xuat()  {  XetTuyen::Xuat();  cout<<"Diem uu tien : "<<DiemUuTien;  }  float DiemTT()  {  return XetTuyen::getDiemTrungBinh()+DiemUuTien;  }  };  int main()  {  int n,m;  //  cout<<"Nhap so luong thi sinh binh thuong : ";cin>>n;  XetTuyen \*ts= new XetTuyen[n+1];  for (int i = 1;i<=n;i++)  {  ts[i].Nhap();  }  for (int i = 1;i<=n;i++)  {  ts[i].Xuat();  }  cout<<"Nhap so luong thi sinh uu tien : ";cin>>m;  UuTien \*ut = new UuTien[m+1];  for (int i = 1;i<=m;i++)  {  ut[i].Nhap();  }  for (int i = 1;i<=m;i++)  {  ut[i].Xuat();  }  //  int i =0;  while((i/n)<=0.3)  {  if(ts[i].getxt())  {  ts[i].Xuat();  }  i++;  }  return 0;  } |
|  | a) Thông tin của các thí sinh thi đại học gồm: tên thí sinh, số báo danh, điểm thi ba môn (môn một, môn hai, môn ba). Hãy cài đặt lớp thí sinh với các phương thức: cấu tử mặc định, cấu tử có tham số, toán tử nhập, xuất và hàm tính tổng điểm (= điểm môn một + điểm môn hai + điểm môn ba).  b) Giả sử đối với các thí sinh là con thương binh, liệt sĩ và con em vùng sâu vùng xa thì có thêm điểm ưu tiên và công thức tính tổng điểm là = điểm môn một + điểm môn hai + điểm môn ba + điểm ưu tiên. Xây dựng các lớp trên sử dụng kỹ thuật đa thể và ràng buộc động. Viết chương trình cho phép nhập vào một mảng các thí sinh, in ra tổng điểm của tất cả các thí sinh đó và cho biết thí sinh nào thi đỗ, thi trượt. Điều kiện thi đỗ là không môn nào bị điểm 0 và tổng điểm >= 15.  Code:  #include<iostream>  using namespace std;  class ThiSinh  {  protected:  char Hoten[50];  float Diem1;  float Diem2;  float Diem3;  public :  ThiSinh(){}  void Nhap()  {  fflush(stdin);  cout<<"Nhap ho ten : ";cin.getline(Hoten,50);  cout<<"Diem mon 1 : ";cin>>Diem1;  cout<<"Diem mon 2 : ";cin>>Diem2;  cout<<"Diem mon 3 : ";cin>>Diem3;  }  void Xuat()  {  cout<<"Nhap ho ten : "<<Hoten;  cout<<"Diem mon 1 : "<<Diem1;  cout<<"Diem mon 2 : "<<Diem2;  cout<<"Diem mon 3 : "<<Diem3;  }  float TongDiem()  {  return Diem1 + Diem2 + Diem3;  }  float getDiem1()  {  return Diem1;  }  float getDiem2()  {  return Diem2;  }  float getDiem3()  {  return Diem3;  }  };  class UuTien : public ThiSinh  {  protected:  float DiemUuTien;  public :  UuTien(){}  void Nhap()  {  ThiSinh::Nhap();  cout<<"Diem uu tien : ";cin>>DiemUuTien;  }  void Xuat()  {  ThiSinh::Xuat();  cout<<"Diem uu tien : "<<DiemUuTien;  }  float DiemTT()  {  return ThiSinh::TongDiem()+DiemUuTien;  }  float getDiem1()  {  return ThiSinh::getDiem1();  }  float getDiem2()  {  return ThiSinh::getDiem2();  }  float getDiem3()  {  return ThiSinh::getDiem3();  }  };  int main()  {  int n,m;  //  cout<<"Nhap so luong thi sinh binh thuong : ";cin>>n;  ThiSinh \*ts= new ThiSinh[n+1];  for (int i = 1;i<=n;i++)  {  ts[i].Nhap();  }  for (int i = 1;i<=n;i++)  {  ts[i].Xuat();  }  cout<<"Nhap so luong thi sinh uu tien : ";cin>>m;  UuTien \*ut = new UuTien[n+1];  for (int i = 1;i<=m;i++)  {  ut[i].Nhap();  }  for (int i = 1;i<=m;i++)  {  ut[i].Xuat();  }  //Do  for (int i = 1;i<=n;i++)  {  if(ts[i].getDiem1() !=0 && ts[i].getDiem2() != 0 && ts[i].getDiem1() != 0 &&ts[i].TongDiem() >= 15)  {  ts[i].Xuat();  }  }  for (int i = 1;i<=m;i++)  {  if(ut[i].getDiem1() !=0 && ut[i].getDiem2() != 0 && ut[i].getDiem3() != 0 &&ut[i].DiemTT() >= 15)  {  ut[i].Xuat();  }  }  return 0;  } |
|  | a) Xây dựng lớp cơ sở trừu tượng mặt hàng (mô tả các mặt hàng nói chung) gồm các thuộc tính dữ liệu: tên mặt hàng, mã mặt hàng, giá nhập, giá xuất và phương thức: tính thuế, khởi tạo, nhập, xuất.  b) Giả sử đối với mặt hàng gia dụng nhập khẩu, ô tô nhập khẩu và máy móc nhập khẩu thì thuế nhập khẩu lần lượt là 10%, 30%, 12% của giá nhập. Hãy cài đặt các lớp trên sử dụng kỹ thuật đa thể và ràng buộc động. Viết chương trình cho phép nhập và in ra danh sách n mặt hàng. Cho biết mặt hàng nào có lãi cao nhất.  CODE: |
|  | a) Một bệnh viện quản lý thông tin bệnh nhân nội trú (họ tên, mã hồ sơ, năm sinh, chuẩn đoán bệnh, tên khoa điều trị, mã phòng+giường) và ngoại trú (họ tên, mã hồ sơ, năm sinh, chuẩn đoán bệnh, tên khoa điều trị, đơn thuốc). Xây dựng các lớp phù hợp có sử dụng kỹ thuật thừa kế.  b) Cài đặt các lớp khác (nếu cần) để có thể xây dựng chương trình nhập và in ra danh sách n bệnh nhân gồm cả nội trú và ngoại trú. Sắp xếp danh sách đó theo mã hồ sơ.  CODE : |
|  | a) Cài đặt lớp thẻ thư viện gồm các thông tin mã thẻ, số ngày mượn, phí mượn theo ngày và các phương thức khởi tạo, nhập dữ liệu, xuất dữ liệu, tính phí mượn = số ngày mượn \* phí mượn. Kế thừa lớp thẻ thư viện để xây dựng lớp thẻ ưu tiên có thêm tỉ lệ giảm giá và các phương thức như lớp thẻ thư viện và tính phí mượn = số ngày mượn \* phí mượn \* (1 - tỉ lệ giảm giá).  b) Viết hàm main nhập vào một mảng các thẻ thư viện (loại thường và loại ưu tiên) sau đó:  + In ra danh sách thẻ vừa nhập;  + In ra danh sách tối đa 10% số thẻ có phí mượn cao nhất và số ngày mượn > 10. |
|  | a) Xây dựng lớp công nhân gồm các thuộc tính dữ liệu họ tên, hệ số lương, phụ cấp và các phương thức nhập, xuất, tính lương = hệ số lương \* 1310 + phụ cấp, tính thưởng = 3 \* lương. Kế thừ từ lớp công nhân xây dựng lớp cán bộ với các phương thức như lớp công nhân nhưng có thêm 1 thuộc tính là phụ cấp đặc thù và lương = hệ số lương \* 1310 + phụ cấp + phụ cấp đặc thù, thưởng = 3.5 \* lương.  b) Viết hàm main nhập vào một mảng n công nhân và cán bộ sau đó:  + In ra mảng vừa nhập;  + In ra danh sách tối đa 20% công nhân hoặc cán bộ có lương thấp nhất và hệ số lương < 3 để nhận hỗ trợ. |
|  | a) Xây dựng lớp các thí sinh xét tuyển đại học gồm các thành viên dữ liệu họ tên, điểm trung bình, điểm tiếng Anh với các phương thức nhập, xuất dữ liệu, xét tuyển (trả về true nếu thí sinh trúng tuyển: điểm trung bình >= 7.0 và điểm tiếng Anh >= 400). Kế thừa từ lớp thí sinh để xây dựng lớp thí sinh ưu tiên có thêm 1 trường dữ liệu là điểm ưu tiên với các phương thức như lớp thí sinh và tiêu chí trúng tuyển là điểm trung bình + điểm ưu tiên >= 7.0 và điểm tiếng Anh >= 400.  b) Viết hàm main nhập vào một mảng các thí sinh (bình thường và ưu tiên) đã xây dựng ở câu a sau đó:  + In ra danh sách các thí sinh vừa nhập;  + In ra danh sách các thí sinh trúng tuyển không vượt quá 30% tổng số thí sinh. |
|  | a) Thế nào là lớp bản mẫu? Điều kiện để sử dụng một lớp dữ liệu cụ thể với một lớp bản mẫu là gì?  b) Khi nào hàm tĩnh được sử dụng, cho ví dụ?  a ,Điều kiện :  - Khi muôn truy xuất vào dữ liệu lớp để thực thi những tác vụ nào đó .  - Có thể tạo bản mẫu để phục vụ cho các tác vụ biến bên ngoài .  b.  Khi chúng ta muốn gọi biến từ bên trong lớp nào đó thì ta sẽ sử dung static .  Từ biến static ta có thể truy xuất ra bên ngoài  VD :  class Time {  public:  void ~Time(); //h?y t?  private:  int h ,p,g;  static int d;  }; |
|  | a) Thế nào là hàm toán tử gọi hàm, cho ví dụ?  b) Hãy cài đặt hàm toán tử tăng dạng hậu tố cho lớp ma trận các số nguyên kích thước 2 x 2.  A,  C++ cho phép bạn xác định nhiều hơn một định nghĩa cho một tên hàm hoặc một toán tử trong cùng phạm vi (scope), được gọi tương ứng là **Nạp chồng hàm (function overloading)** và **Nạp chồng toán tử (operator overloading)** trong C++.  B, |
|  | - Trình bày sự hiểu biết của em về thành phần static trong Lập trình HĐT;  - Hãy giải thích và sửa lại những chỗ sai trong khai báo sau:  class Time {  public:  void ~Time(int); //hủy tử  private:  int h = 0; m = 0; n = 0;  static int d = 0;  };  Giải thích :  int h = 0; m = 0; n = 0;  static int d = 0;   * Bởi vì tham số khởi tạo không được truyện trực tiếp như vậy . Phải khởi tạo bên dưới public .   void ~Time(int);  -Hàm hủy không có biến truyền vào  Sửa :  int h , m , n ;  static int d ;  void ~Time(){ } |
|  | - Trình bày sự hiểubiết của bạn về bản mẫu hàm.  - Xây dựng bản mẫu hàm tMax để có thể tìm được vị trí phần tử lớn nhất từ phần tử thứ N đến phần tử thứ M trong mảng A.  Template là nền tảng của lập trình tổng quát (generic programming), tức là viết code theo các mà độc lập với bất kỳ kiểu cụ thể nào.  Một Template là một blueprint hoặc là phương thức để tạo một lớp hoặc một hàm tổng quát. Các Library Container như Iterator và các thuật toán là các ví dụ của lập trình tổng quát và đã được phát triển bởi sử dụng khái niệm Template.  Mỗi container có một định nghĩa đơn, ví dụ vector, nhưng chúng ta có thể định nghĩa nhiều loạt vector khác nhau, ví dụ: **vector <int>** hoặc **vector <string>**.  Template là từ khóa trong C++, chúng ta có thể hiểu rằng là nó một kiểu dữ liệu trừu tượng, đặc trưng cho các kiểu dữ liệu cơ bản. Template là từ khóa báo cho trình biên dịch rằng đoạn mã sau đây định nghĩa cho nhiều kiểu dữ liệu và mã nguồn của nó sẽ được biên dịch sinh ra tương ứng cho từng kiểu dữ liệu trong quá trình biên dịch. Có hai kiểu Template trong C++:   * Function Template: là một khuôn mẫu hàm, cho phép định nghĩa các hàm tổng quát thao tác cho nhiều kiểu dữ liệu. * Class template: là một khuôn mẫu lớp, cho phép định nghĩa các lớp tổng quát cho nhiều kiểu dữ liệu   --  #include <iostream>  using namespace std;  template<typename T>  int timmax(T &mang, T &N , T &M)  {  int max=0;  for (int i=N;i<M;i++)  {  if(max < mang[i])  {  max = mang[i];  }  }  return max;  } |
|  | - Trình bày sự hiểu biết của bạn về quá tải toán tử. Nêu ví dụ về toán tử ++, --  - Bạn hãy sửa đoạn chương trình sau nếu có lỗi và cho biết kết quả hiển thị sau khi thực hiện hàm main().  class A {  private: int x,y;  public:  A(int x1, int y1) { x=x1; y=y1; } //cấu tử  void operator+=(A a) { x+=a.x; y+=a.y; }  void operator=+(A a) { x+=a.x; y+=a.y; }  void In() { cout << x << “, ”; cout << y; }  };  Int main() {  A a1(5,5);  A a2(6,6);  a2+=a1;  a1=+a2;  a2.In(); a1.In();  }  Sai :  void operator=+(A a) { x+=a.x; y+=a.y; }  Sửa :  void operator=+(A a) { x=+a.x; y=+a.y; }  Kết quả :  a2+=a1;// Cong truoc ms gan  //a2 (11,11)  a1=+a2;// Gan truoc ms cong  //a1(12,12)  a2.In(); a1.In(); |
|  | a) Thế nào là toán tử gọi hàm, cho ví dụ?  b) Sửa lỗi trong đoạn chương trình sau:  class B {  int id; // sai  public:  static int getID () { return id;};  static int num = 0;// sai  };  Sửa :  class B {  static int id;  public:  static int getID () { return id;}  static int num;  }; |
|  | a) Trình bày các hiểu biết của em về lớp trừu tượng, cho ví dụ?  b) Khai báo và cài đặt hàm bản mẫu sắp xếp một mảng các đối tượng.  a,  Một Interface miêu tả hành vi hoặc khả năng của một lớp trong C++ mà không ký thác tới một trình triển khai cụ thể của lớp đó.  Interface trong C++ được triển khai bởi sử dụng các **Lớp trừu tượng (Abstract class)** và những lớp trừu tượng này không nên bị nhầm lẫn với Trừu tượng hóa dữ liệu, mà là một khái niệm của việc giữ Implementation Detail phân biệt với dữ liệu được liên kết.  B,  #include <iostream>  using namespace std;  template<typename T>  int timmax(T &a ,T &n)  {  for (int i=0;i<n;i++)  {  for (int j=1;j<n;j++)  {  if(a[i] < a[j])  {  int temp = a[i];  a[i] = a[j];  a[j] = temp;  }  }  }  } |
|  | a) Trình bày các hiểu biết của em về hàm ảo thực sự?  b) Khai báo, cài đặt và sử dụng một hàm tĩnh tính độ dài của một đoạn thẳng biết tọa độ điểm đầu và điểm cuối.  a,  b,  static float DoDai(int x1,int x2,int y1,int y2)  {  return sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1));  } |