Lưu ý sinh viên Save As lại file word này và nộp lại cho GV:

* File .cpp của các bài tập

1. **Xây dựng lớp SanPham\_HoaDon được mô tả như sau:**

Lập chương trình thực hiện các công việc sau:

**Xây dựng lớp cơ sở sản phẩm gồm:**

* Thuộc tính: Tên sản phẩm, năm sản xuất, giá thành
* Phương thức: Nhập, xuất thông tin

**Xây dựng lớp hoá đơn bán sản phẩm kế thừa từ lớp sản phẩm có thêm:**

* Thuộc tính: Số lượng bán, giá bán
* Phương thức: Nhập, xuất thông tin, tính thành tiền (=số lượng \* giá bán), tính thuế (=10% thành tiền), tính lãi (chênh lệch giá bán \* số lượng bán).

**Chương trình chính thực hiện:**

* Nhập danh sách N hoá đơn bán sản phẩm
* Sắp xếp danh sách theo tiền lãi giảm dần
* Hiện ra màn hình danh sách gồm: số thứ tự, tên sản phẩm, giá thành, số lượng bán, giá bán, thành tiền, thuế và tiền lãi.
* Tính tổng tiền của các hoá đơn bán sản phẩm
* Cho biết thông tin các hoá đơn bán sản phẩm có tiền thuế cao nhất

Lưu bài với định dạng: *Bai28\_* **SanPham\_HoaDon** *\_MSSV\_HoTen.cpp*

YÊU CÂU 1: Sinh viên copy đoạn mã code đã thiết kế để ở đây:

|  |
| --- |
| /\*  MSSV:223492  LOP: DH22TIN04  HO VA TEN: HUYNH VAN NHAN  \*/  #include <iostream>  #include <string>  #include <algorithm>  using namespace std;  class SanPham\_HoaDon {  private:  string TenSanPham;  string NamSanXuat;  long GiaThanh;  public:  SanPham\_HoaDon(string TenSanPham = "",string NamSanXuat = "", long GiaThanh = 0);  ~SanPham\_HoaDon();  void Input();  void Output();  long getGiaThanh() const {  return GiaThanh;  }  };  class HoaDon\_BanSanPham : public SanPham\_HoaDon {  private:  int SoLuongBan;  long GiaBan;  public:  void Input();  void Output();  long TinhTien();  long TinhTienThue();  long TinhTienLai();  };  SanPham\_HoaDon::SanPham\_HoaDon (string TenSanPham, string NamSanXuat,long GiaThanh) : TenSanPham(TenSanPham), NamSanXuat(NamSanXuat), GiaThanh(GiaThanh) {}  SanPham\_HoaDon::~SanPham\_HoaDon() {}  void SanPham\_HoaDon::Input() {  cout << "===========NHAP THONG TIN HOA DON BAN SAN PHAM===========" << endl;  cout << "\n NHAP TEN SAN PHAM: ";  getline(cin, TenSanPham);  cout << "\n NHAP NAM SAN XUAT: ";  getline(cin, NamSanXuat);  cout << "\n NHAP GIA THANH: ";  cin >> GiaThanh;  }  void SanPham\_HoaDon::Output() {  cout << "\n===========THONG TIN HOA DON BAN SAN PHAM DA NHAP===========" << endl;  cout << "\nTEN SAN PHAM: " << TenSanPham <<"\t NAM SAN XUAT: " << NamSanXuat << "\t GIA THANH: " << GiaThanh << endl;  }  void HoaDon\_BanSanPham::Input() {  SanPham\_HoaDon::Input();  cout << "\n NHAP SO LUONG BAN: ";  cin >> SoLuongBan;  cout << "\n NHAP GIA BAN: ";  cin >> GiaBan;  cin.ignore();  }  void HoaDon\_BanSanPham::Output() {  SanPham\_HoaDon::Output();  cout << "\n\t SO LUONG BAN: " << SoLuongBan << "\t GIA BAN: "<< GiaBan << endl;  cout << "\n \t THANH TIEN: " << TinhTien() << "\t TIEN THUE: " << TinhTienThue() << "\t TIEN LAI: " << TinhTienLai() << endl;  }  long HoaDon\_BanSanPham::TinhTien() {  long TinhThanhTien;  TinhThanhTien = SoLuongBan \* GiaBan;  return TinhThanhTien;  }  long HoaDon\_BanSanPham::TinhTienThue() {  long TinhThue;  TinhThue = (TinhTien()/100)\*10;  return TinhThue;  }  long HoaDon\_BanSanPham::TinhTienLai() {  long TinhLai;  TinhLai = (GiaBan - getGiaThanh()) \* SoLuongBan;  return TinhLai;  }  // HOAN DOI VI TRI HAI PHAN TU  void swap(HoaDon\_BanSanPham\* a, HoaDon\_BanSanPham\* b) {  HoaDon\_BanSanPham t = \*a;  \*a = \*b;  \*b = t;  }  // HAM PHAN VUNG DUNG TRONG QuickSort  int partition (HoaDon\_BanSanPham arr[], int low, int high) {  long pivot = arr[high].TinhTienLai(); // pivot  int i = (low - 1); // CHI SO CUA PHAN TU NHO HON  for (int j = low; j <= high - 1; j++) {  // NEU PHAN TU HIEN TAI LON HON HOAC BANG pivot  if (arr[j].TinhTienLai() >= pivot) {  i++; // TANG CHI SO CUA PHAN TU NHO HON  swap(&arr[i], &arr[j]);  }  }  swap(&arr[i + 1], &arr[high]);  return (i + 1);  }  // HAM THUC HIEN THUAT TOAN QuickSort  void quickSort(HoaDon\_BanSanPham arr[], int low, int high) {  if (low < high) {  // pi LA CHI SO PHAN VUNG, arr[p] DA DUNG VI TRI  int pi = partition(arr, low, high);  // SAP XEP DE QUY CAC PHAN TU TRUOC VA SAU PHAN VUNG  quickSort(arr, low, pi - 1);  quickSort(arr, pi + 1, high);  }  }  int main() {  int n;  cout << "\t NHAP SO HOA DON BAN SAN PHAM N= ";  cin >> n;  cin.ignore();  HoaDon\_BanSanPham a[n];  for (int i = 0; i < n; i++) {  cout << "\nHOA DON: " << (i + 1) << ": \n";  a[i].Input();  }  // SAP XEP MANG THEO THU TU GIAM DAN CUA LOI NHUAN  quickSort(a, 0, n-1);  // XUAT MANG DA SAP XEP  cout << "\nDanh sach sau khi sap xep giam dan theo tien lai:\n";  for (int i = 0; i < n; i++) {  cout << "\nHoa don: " << (i + 1) << ": \n";  a[i].Output();  }  // TIM HOA DON CO TIEN THUE CAO NHAT  long maxTax = a[0].TinhTienThue();  for (int i = 1; i < n; i++) {  long tax = a[i].TinhTienThue();  if (tax > maxTax) {  maxTax = tax;  }  }  // XUAT THONG TIN CAC HOA DON CO TIEN THUE CAO NHAT  cout << "\nTHONG TIN CAC HOA DON CO TIEN THUE CAO NHAT:\n";  for (int i = 0; i < n; i++) {  if (a[i].TinhTienThue() == maxTax) {  cout << "\nHoa don: " << (i + 1) << ": \n";  a[i].Output();  }  }  // TINH TONG TIEN CUA TAT CA CAC DON HANG  long long total = 0;  for (int i = 0; i < n; i++) {  total += a[i].TinhTien();  }  // XUAT TONG TIEN  cout << "\nTONG TIEN CUA TAT CA DON HANG: " << total << endl;  return 0;  } |

YÊU CÂU 2: Sinh viên chụp kết quả chạy chương trình để ở đây

|  |
| --- |
|  |

1. **Xây dựng lớp CanBo\_GiangVien được mô tả như sau:**

Trong một trường Đại học cần quản lý cán bộ và giảng viên.

Cán bộ gồm các thông tin: Mã cán bộ, mã đơn vị, năm sinh, hệ số lương, phụ cấp ăn ca, bảo hiểm.

Giảng viên cần bổ sung thêm các thông tin: phụ cấp đứng lớp =25% lương cơ bản và phụ cấp độc hại (nếu có) = 10% lương cơ bản.

Hãy lập chương trình quản lý cán bộ và giảng viên sao cho kế thừa được các dữ liệu dùng chung thực hiện tính lương cho cán bộ và giảng viên tương ứng, in ra danh sách cán bộ và giảng viên phải đóng thuế thu nhập (lương thực lĩnh>5000000).

Lưu bài với định dạng: *Bai29\_* **CanBo\_GiangVien** *\_MSSV\_HoTen.cpp*

YÊU CÂU 1: Sinh viên copy đoạn mã code đã thiết kế để ở đây:

|  |
| --- |
| /\*  MSSV:223492  LOP: DH22TIN04  HO VA TEN: HUYNH VAN NHAN  \*/  #include <iostream>  #include <string>  using namespace std;  class Can\_Bo {  protected:  string Ma\_CanBo;  string Ma\_DonVi;  string Nam\_Sinh;  float HeSoLuong;  long PhuCapAn;  long BaoHiem;  long Luong1;  public:  Can\_Bo(string Ma\_CanBo = "",string Ma\_DonVi = "", string Nam\_Sinh = "", float HeSoLuong = 0, long PhuCapAn = 0, long BaoHiem = 0);  ~Can\_Bo();  void Input();  void Output();  long Luong();  };  class Giang\_Vien : public Can\_Bo {  private:  int DH;  public:  void Input();  void Output();  };  Can\_Bo::Can\_Bo (string Ma\_CanBo, string Ma\_DonVi, string Nam\_Sinh, float HeSoLuong, long PhuCapAn, long BaoHiem) : Ma\_CanBo(Ma\_CanBo), Ma\_DonVi(Ma\_DonVi), Nam\_Sinh(Nam\_Sinh) ,HeSoLuong(HeSoLuong),PhuCapAn(PhuCapAn),BaoHiem(BaoHiem){}  Can\_Bo::~Can\_Bo() {}  void Can\_Bo::Input() {  cout << "===========NHAP THONG TIN CAN BO===========" << endl;  cout << "\n NHAP MA CAN BO: ";  getline(cin, Ma\_CanBo);  cout << "\n NHAP MA DON VI: ";  getline(cin, Ma\_DonVi);  cout << "\n NHAP NAM SINH: ";  getline(cin, Nam\_Sinh);    cout << "\n NHAP HE SO LUONG: ";  cin >> HeSoLuong;  cout << "\n NHAP PHU CAP AN: ";  cin >> PhuCapAn;  cout << "\n NHAP BAO HIEM: ";  cin >> BaoHiem;    }  Long Can\_Bo::Luong(){  Long LuongCS = 1500000;  Luong1 = LuongCS \* HeSoLuong + PhuCapAn + ((1500000/100)\*25) - BaoHiem;  return Luong1;  }  void Can\_Bo::Output() {  cout << "\n===========THONG TIN CAN BO DA NHAP===========" << endl;  cout << "\nMA CAN BO: " << Ma\_CanBo <<"\t MA DON VI: " << Ma\_DonVi << "\t NAM SINH: " << Nam\_Sinh <<"\t HE SO LUONG: " << HeSoLuong << "\t PHU CAP AN: " << PhuCapAn << "\t BAO HIEM: " << BaoHiem << endl;  cout << "\n LUONG: " << Luong() << endl;  }  void Giang\_Vien::Input(){  Can\_Bo::Input();  cout << "\n NEU NEU CO DOC HAI NHAP 1 , KHONG CO DOC HAI NHAP 0 ";  cin >> DH;  cin.ignore();  }  void Giang\_Vien::Output(){  Can\_Bo::Output();  if (DH == 1)  cout << "\n TONG LUONG CUA BAN LA: " << Luong1 + ((1500000/100)\*10);  else  cout << "\n TONG LUONG CUA BAN LA: " << Luong1;  }  int main() {  int n;  cout << "\t NHAP SO GIANG VIEN N= ";  cin >> n;  cin.ignore();  Giang\_Vien\* a = new Giang\_Vien[n];  for (int i = 0; i < n; i++) {  cout << "\nGIANG VIEN THU: " << (i + 1) << ": \n";  a[i].Input();  }  cout << "\n THONG TIN VUA NHAP: \n";  for (int i = 0; i < n; i++) {  cout << "\nGIANG VIEN THU: " << (i + 1) << ": \n";  a[i].Output();  }  delete[] a;  return 0;  } |

YÊU CÂU 2: Sinh viên chụp kết quả chạy chương trình để ở đây

|  |
| --- |
|  |

1. **Xây dựng lớp NhanVien\_KySu được mô tả như sau:**

Lập chương trình thực hiện các việc sau:

Tạo một lớp **nhân viên** với:

* Dữ liệu gồm: Tên nhân viên, Đơn vị, Hệ số lương, lương tối thiểu với
* Các phương gồm:
  + Nhập dữ liệu
  + Tính lương theo công thức: Lương chính = Lương tối thiểu \* Hệ số lương.

Hiện tại có nhu cầu tính lương mới theo cách sau:

* Kỹ sư: Lương mới = Lương chính + Số năm trong nghề \* Phụ cấp chuyên môn
* Các nhân viên khác vẫn tính lương theo cách tính như trước

Hãy thiết kế chương trình thực hiện việc tính lương mới cho các nhân viên bằng phương pháp kế thừa chương trình đã có trước.

Lưu bài với định dạng: *Bai30\_* **NhanVien\_KySu** *\_MSSV\_HoTen.cpp*

YÊU CÂU 1: Sinh viên copy đoạn mã code đã thiết kế để ở đây:

|  |
| --- |
| /\*  MSSV:223492  LOP: DH22TIN04  HO VA TEN: HUYNH VAN NHAN  \*/  #include <iostream>  #include <string>  using namespace std;  class NhanVien {  private:  string TenNhanVien;  string DonVi;  float HeSoLuong;  long LuongToiThieu;  long LuongChinh;  long getLuongChinh() const {  return LuongChinh;  }  public:  NhanVien(string TenNhanVien = "",string DonVi = "", float HeSoLuong = 0, long LuongToiThieu = 0);  ~NhanVien();  void Input();  void Output();  long Luong();  };  class KySu : public NhanVien {  private:  int ks;  int SoNam;  long PhuCapChuyenMon;  public:  void Input();  void Output();  };  NhanVien::NhanVien(string TenNhanVien, string DonVi, float HeSoLuong, long LuongToiThieu) : TenNhanVien(TenNhanVien), DonVi(DonVi), HeSoLuong(HeSoLuong) ,LuongToiThieu(LuongToiThieu) {}  NhanVien::~NhanVien() {}  void NhanVien::Input() {  cout << "===========NHAP THONG TIN NHAN VIEN===========" << endl;  cout << "\n NHAP TEN NHAN VIEN: ";  getline(cin, TenNhanVien);  cout << "\n NHAP DON VI: ";  getline(cin, DonVi);  cout << "\n NHAP HE SO LUONG: ";  cin >> HeSoLuong;  cout << "\n NHAP LUONG TOI THIEU: ";  cin >> LuongToiThieu;  cin.ignore();  }  long NhanVien::Luong() {  LuongChinh = LuongToiThieu \* HeSoLuong;  return LuongChinh;  }  void NhanVien::Output() {  cout << "\n===========THONG TIN NHAN VIEN DA NHAP===========" << endl;  cout << "\nTEN NHAN VIEN: " << TenNhanVien <<"\t DON VI: " << DonVi <<"\t HE SO LUONG: " << HeSoLuong << endl;  cout << "\n LUONG CHINH: " << Luong() << endl;  }  void KySu::Input() {  NhanVien::Input();  cout << "\n NEU LA KY SU NHAP 1 , KHONG KHONG LA KY SU NHAP 0:";  cin >> ks;  cout << "\n SO NAM TRONG NGHE CUA BAN LA: ";  cin >> SoNam;  cout << "\n PHU CAP CHUYEN MON:";  cin >> PhuCapChuyenMon;  cin.ignore();  }  void KySu::Output() {  NhanVien::Output();  if (ks == 1) {  cout << "\n TONG LUONG CUA BAN LA: " << Luong() + SoNam \* PhuCapChuyenMon ;  } else  cout << "\n TONG LUONG CUA BAN LA: " << Luong();  }  int main() {  int n;  cout << "\t NHAP SO NHAN VIEN N= ";  cin >> n;  cin.ignore();  KySu\* a = new KySu[n];  for (int i = 0; i < n; i++) {  cout << "\nNhanVien THU: " << (i + 1) << ": \n";  a[i].Input();  }  cout << "\n THONG TIN VUA NHAP: \n";  for (int i = 0; i < n; i++) {  cout << "\nNhanVien THU: " << (i + 1) << ": \n";  a[i].Output();  }  delete[] a;  return 0;  } |

YÊU CÂU 2: Sinh viên chụp kết quả chạy chương trình để ở đây

|  |
| --- |
|  |