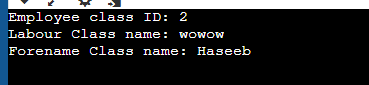
**Task 1:**

**Implement the given UML, you have to identify the appropriate attributes of each class and you have to write the getter setter methods, default parameterized constructor, set method and display method of each class**

|  |
| --- |
| #include <iostream>  using namespace std;  class employee  {  protected:  int id;  public:  employee() {}  employee(int i): id(i){}  void display()  {  cout << "Employee class ID: "<< id << endl;  }  };  class manage : public employee  {  int name;  public:  manage(){}  void display()  {  cout << "Manage name: "<< name << endl;  }  };  class Scientist : public employee  {  int s\_name;  public:  Scientist(){}  void display()  {  cout << "Scientist name: "<< s\_name << endl;  }  };  class Labour : public employee  {  string Labour\_name;  public:  Labour(int id,string lname): Labour\_name(lname), employee(id) {}    void display()  {  cout << "Labour Class name: "<< Labour\_name << endl;  }  };  class forename : public Labour  {  string fore\_name;  public:  forename(string for\_nam, string lname, int id ): fore\_name(for\_nam), Labour(id , lname) {}    void display()  {  employee::display();  Labour::display();  cout << "Forename Class name: "<< fore\_name << endl;    }  };  int main()  {  forename a("Haseeb", "wowow", 2);  a.display();  return 0;  } |

Output



**Task 2:**

**Implement the given UML, you have to identify the appropriate attributes of each class and you have to write the getter setter methods, default parameterized constructor, set method and display method of each class**

|  |
| --- |
| #include <iostream>  using namespace std;  class person  {  protected:  string name;  public:  person() {}  person(string n): name(n) {}  void display()  {  cout << "person: " << name <<endl;  }  };  class Employee : public virtual person  {  protected:  int id;  public:  Employee() {}  Employee(int i,string n): id(i),person(n) {}  void display()  {  cout << "Employee: " << id <<endl;  }  };  class student : public virtual person  {  protected:  string university;  public:  student() {}  student(string uni,string na): university(uni),person(na) {}  void display()  {  cout << "Student: " << university <<endl;  }  };  class manage : public person, public student  {  protected:  int salary;  public:  manage(){}  manage(int sal, string n,string uni) : salary(sal), person(n), student(uni) {}  void display()  {  cout << " Manage Salary: " << salary <<endl;  }  }  class scientist : public Employee, public student  {  protected:  string s\_name;  public:  scientist(){}  scientist(string sci, string n,string uni) : s\_name(sci), person(n), student(uni) {}  void display()  {  cout << "Scientist: " << s\_name <<endl;  }  }  class Labour : public Employee  {  protected:  int hours;  public:  Labour(){}  Labour(int h, int i, string n ) : hours(h), Employee(i,n) {}  void display()  {  cout << "Hour: " << hours <<endl;  }  };  int main()  {  Labour l;  return 0;  } |

Output

