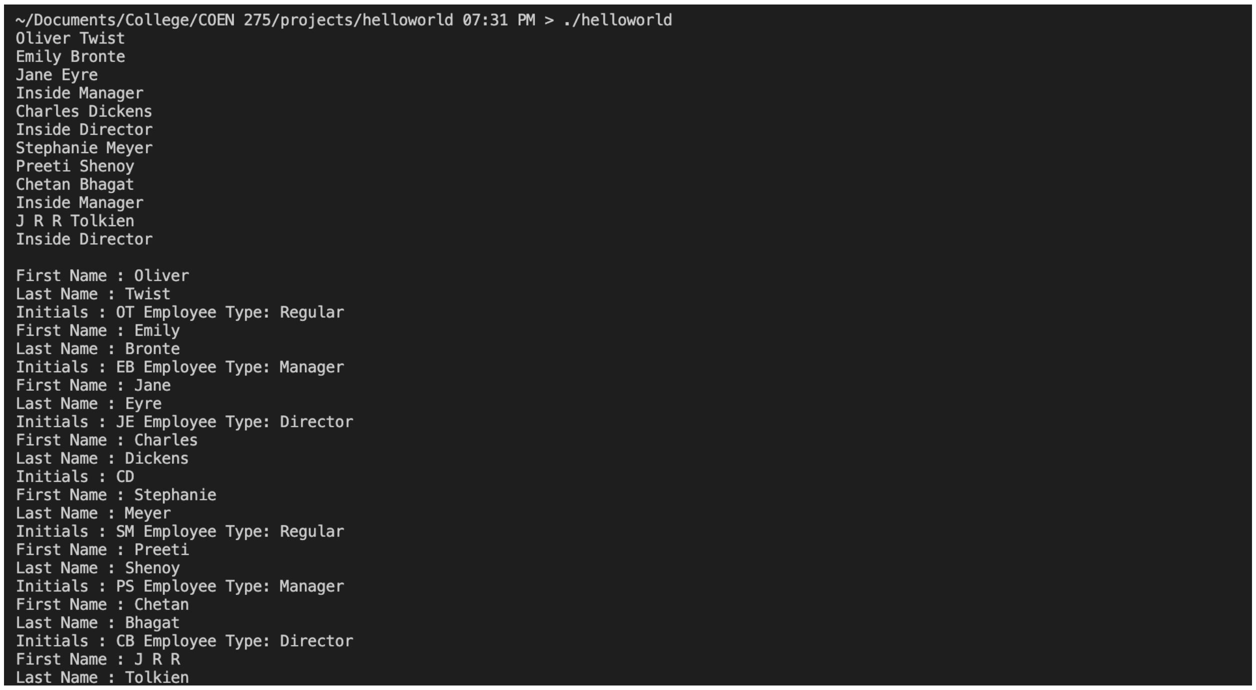
Assignment 2

```cpp
#include <iostream>
#include <vector>
using namespace std;

class Person {
   private:
    string fname;
    string lname;
    string initials;

   public:
   Person(){}
   //***** Answer to Q1 overridden by Q2 **********
   // Person(string fn, string ln) {
   //      fname = fn;
   //      lname = ln;
   //      setInitalsFromName();
   //      cout << fn << " " + ln << "  " + initials << endl;
   // }

   //***** Answer to Q2 **********
   Person(string fn, string ln): Person(fn, ln, setInitalsFromName(fn, ln)) {
       // calling 3 parameterized constructor above
   }
   string setInitalsFromName(string fn, string ln) {
       std::string str = std::string() + fn.at(0) + ln.at(0);
       return str;
   }
   Person(string fn, string ln, string it) {
       fname = fn;
       lname = ln;
       initials = it;
       cout << fn << " " +ln <<endl;
   }
   Person( const Person &obj) {
       fname = obj.fname;
       lname = obj.lname;
       setInitalsFromName();
   };
   Person& operator=(const Person& t)
   {
       cout << "Assignment operator " << endl;
       return *this;
   }
   ~Person(){};

   void setInitals(string it) {
       initials = it;
   }

   void setInitalsFromName() {
       std::string str = std::string() + fname.at(0) + lname.at(0);
       initials = str;
   }

   void setFname(string name) {
       fname = name;
   }

   void setLname(string name) {
       lname = name;
   }

   string getFname() {
       return fname;
   }
   string getLname() {
       return lname;
   }
   string getInitials() {
       return initials;
   }

   //***** Answer to Q4 **********
   virtual string toString() {
       string str = "\nFirst Name : " + fname + " \nLast Name : " + lname + "
\nInitials : " + initials;
       return str;
   }
};
```

```
//***** Answer to Q3 **********
class Employee: public Person {
 public:
   string employeeId;
    Employee()
   {
       cout << "Inside Employee" << endl;
   }
    Employee(string fn, string ln, string empId): Person(fn, ln)
   {
        employeeId = empId;
   }
   void setEmployeeId(string empId) {
       employeeId = empId;
   }
   //***** Answer to Q4 **********
   string toString() {
       string str = Person::toString();
       return " Employee Type: Regular " + str;
   }
};

class Manager: public Employee {
   public:
   Manager()
   {
       cout << "Inside Manager" << endl;
   }
   Manager(string fn, string ln, string empId): Employee(fn, ln, empId)
   {
       cout << "Inside Manager" << endl;
   }
   //***** Answer to Q4 **********
   string toString() {
       string str = Person::toString();
       return " Employee Type: Manager " + str;
   }
};

class Director: public Employee {
   public:
   Director()
```

```
    {
        cout << "Inside Director" << endl;
    }

    Director(string fn, string ln, string empId): Employee(fn, ln, empId)
    {
        cout << "Inside Director" << endl;
    }
    string toString() {
        string str = Person::toString();
        return " Employee Type: Director " + str;
    }
};
//***** Answer to Q5 **********
int main() {
    std::vector <Person*> people;

    Person* person = new Person("Oliver","Twist");
    Employee* employee = new Employee("Emily","Bronte", "emp001");
    Manager* manager = new Manager("Jane", "Eyre", "emp002");
    Director* director = new Director("Charles", "Dickens", "emp003");

    Person* person2 = new Person("Stephanie","Meyer");
    Employee* employee2 = new Employee("Preeti","Shenoy", "emp004");
    Manager* manager2 = new Manager("Chetan", "Bhagat", "emp005");
    Director* director2 = new Director("J R R", "Tolkien", "emp006");

    people.push_back(person);
    people.push_back(employee);
    people.push_back(manager);
    people.push_back(director);
    people.push_back(person2);
    people.push_back(employee2);
    people.push_back(manager2);
    people.push_back(director2);


    for (int i = 0; i < (int)people.size(); i++)
        cout<< people.at(i)->toString();

    return 0;
}
```

Output :



```
~/Documents/College/COEN 275/projects/helloworld 07:31 PM > ./helloworld
Oliver Twist
Emily Bronte
Jane Eyre
Inside Manager
Charles Dickens
Inside Director
Stephanie Meyer
Preeti Shenoy
Chetan Bhagat
Inside Manager
J R R Tolkien
Inside Director

First Name : Oliver
Last Name : Twist
Initials : OT Employee Type: Regular
First Name : Emily
Last Name : Bronte
Initials : EB Employee Type: Manager
First Name : Jane
Last Name : Eyre
Initials : JE Employee Type: Director
First Name : Charles
Last Name : Dickens
Initials : CD
First Name : Stephanie
Last Name : Meyer
Initials : SM Employee Type: Regular
First Name : Preeti
Last Name : Shenoy
Initials : PS Employee Type: Manager
First Name : Chetan
Last Name : Bhagat
Initials : CB Employee Type: Director
First Name : J R R
Last Name : Tolkien
```

Oliver Twist
Emily Bronte
Jane Eyre
Inside Manager
Charles Dickens
Inside Director
Stephanie Meyer
Preeti Shenoy
Chetan Bhagat
Inside Manager
J R R Tolkien
Inside Director

First Name : Oliver
Last Name : Twist
Initials : OT Employee Type: Regular
First Name : Emily
Last Name : Bronte
Initials : EB Employee Type: Manager
First Name : Jane

```
Last Name : Eyre
Initials : JE Employee Type: Director
First Name : Charles
Last Name : Dickens
Initials : CD
First Name : Stephanie
Last Name : Meyer
Initials : SM Employee Type: Regular
First Name : Preeti
Last Name : Shenoy
Initials : PS Employee Type: Manager
First Name : Chetan
Last Name : Bhagat
Initials : CB Employee Type: Director
First Name : J R R
Last Name : Tolkien
```