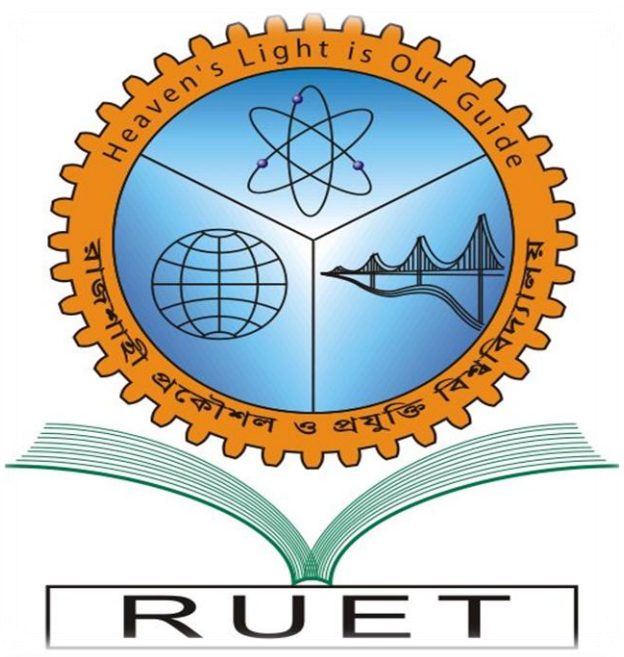


Lab 4

Lab Report on CSE 1204

Submitted by

Name: Md. Shabir Khan Akash.

Class: 1st year, even semester.

Dept.: Department of CSE.

Roll: 1603108

Section: B

Submitted to

Shyla Afroge Madam

Assistant Professor

Department of CSE, RUET

Theory

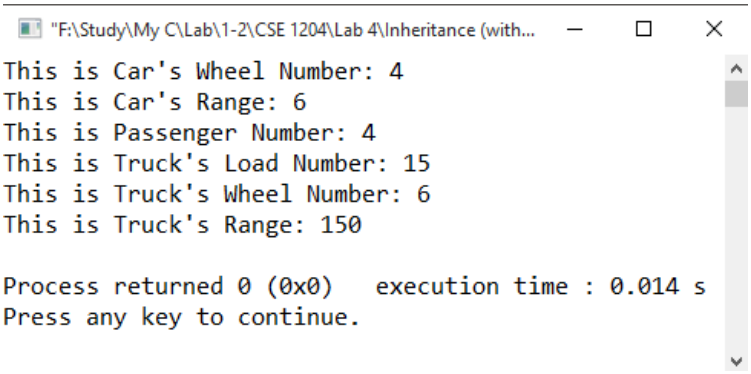
Inheritance is the process by which one object can acquire the properties of another specifically, an object can inherit a general set of to properties to which it can add those features that are specific only to itself. Inheritance is important because it allows an object to support the concept of *hierarchical classification*. So. Inheritance certainly plays a very important role in OOP.

Code

(With constructor)

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

class vehicle {
public:
   int wheel_no;
   int range;

   vehicle (int w, int r) {
      wheel_no = w;
      range = r;
   }
   void show() {
        cout<<"This is Car's Wheel Number: "<<wheel_no<<endl;
        cout<<"This is Car's Range: "<<range<<endl;
   }
};
   class car : public vehicle {
      int passenger;
   public:
      car(int p, int w, int r) : vehicle(w,r) {
         passenger = p;
      }
      void show_passenger() {
         cout<<"This is Passenger Number: "<<passenger<<endl;
      }
   };

      class truck : public vehicle {
         public:
         int loadNumber;
         truck (int l, int w, int r) : vehicle(w,r) {
            loadNumber = l;
         }
      void show() {
         cout<<"This is Truck's Load Number: "<<loadNumber<<endl;
         cout<<"This is Truck's Wheel Number: "<<wheel_no<<endl;
         cout<<"This is Truck's Range: "<<range<<endl;      }
      };


int main()
{
   car c(4,4,6);
   truck t(15,6,150);
   c.show();
   c.show_passenger();
   t.show();
   return 0;
}
```

Output



Comment

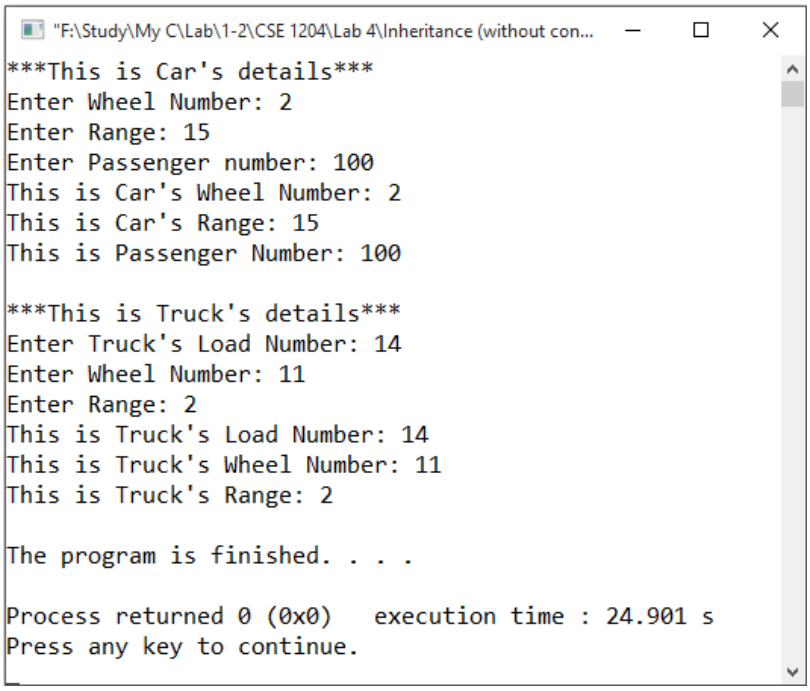
In this above code, a class named ***vehicle*** was declared where two integer variables ***wheel_no,*** and ***range*** was declared in public and also a constructor was declared to take the values of the two above variable from the initializing object. After that, a derived class of ***vehicle*** class, named ***car*** was declared where a constructor was declared to get the ***passenger*** number of the car. As it was derived from class ***vehicle*** so it has the access to the member of ***vehicle*** class. Then another derived class of ***vehicle*** was declared and it was ***truck*** where another constructor was declared to get the ***loadNumber*** of the ***truck.*** Finally in the main function, a ***car*** type object ***c*** and a ***truck*** type object ***t*** was declared and also initialized with values. Then the ***show()*** function was called for both ***car*** and ***truck*** class and ***show_passenger()*** was called to show the number of passenger.

Code

(Without Constructor)

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

class vehicle {
public:
  int wheel_no;
  int range;
  void get_wheel_no() {
    cout<<"Enter Wheel Number: ";
    cin>>wheel_no;
  }
  void get_range() {
    cout<<"Enter Range: ";
    cin>>range;
  }
  void show() {
      cout<<"This is Car's Wheel Number: "<<wheel_no<<endl;
      cout<<"This is Car's Range: "<<range<<endl;
  }
};
  class car : public vehicle {
  public:
    int passenger;
    void get_passenger() {
      cout<<"Enter Passenger number: ";
      cin>>passenger;
    }
    void show_passenger() {
      cout<<"This is Passenger Number: "<<passenger<<endl;
    }
  };

    class truck : public vehicle {
      public:
      int loadNumber;
    void get_load_number() {
      cout<<"Enter Truck's Load Number: ";
      cin>>loadNumber;
    }
    void show() {
      cout<<"This is Truck's Load Number: "<<loadNumber<<endl;
      cout<<"This is Truck's Wheel Number: "<<wheel_no<<endl;
      cout<<"This is Truck's Range: "<<range<<endl;     }
    };

int main()
{
  car c;
  truck t;
  cout<<"***This is Car's details***"<<endl;
  c.get_wheel_no();
  c.get_range();
  c.get_passenger();
  c.show();
  c.show_passenger();
  cout<<'\n';
  cout<<"***This is Truck's details***"<<endl;
  t.get_load_number();
  t.get_wheel_no();
  t.get_range();
  t.show();
  cout<<"\nThe program is finished. . . ."<<endl;

  return 0;
}
```

Output



Comment

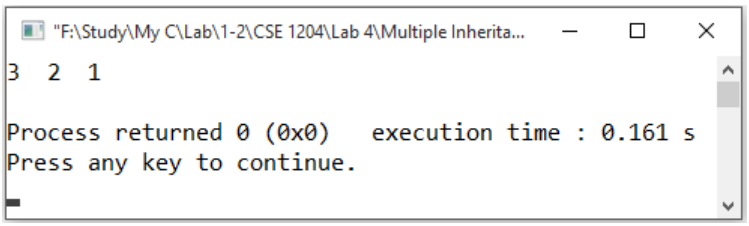
In this above code, a class named ***vehicle*** was declared where two integer variables ***wheel_no,*** and ***range*** was declared in public and two functions ***get_wheel_no()*** and ***get_range()*** was declared to take the values of the two above variable for the object. After that, a derived class of ***vehicle*** class, named ***car*** was declared where two functions ***get_passenger()*** and ***show_passenger()*** was declared to get the passenger number of the car from the console and show the passenger number respectively. As it was derived from class ***vehicle*** so it has the access to the member of ***vehicle*** class. Then another derived class of ***vehicle*** was declared and it was ***truck*** where a function ***get_loadNumber()*** was declared to get the ***loadNumber*** of the ***truck*** from the console**.** Then ***show()*** function was declared to show the wheel number, range and load-number of the truck. Finally in the main function, a ***car*** type object ***c*** and a ***truck*** type object ***t*** was declared and also initialized with values. Then all the functions were called for both ***car*** and ***truck*** class to show the above output.

Code

(Multiple Inheritance)

```cpp
#include<iostream>
using namespace std;
class A {
  int a;
public:
  A (int x) { a = x; }
  int get_a() { return a; }
  };

class B {
  int b;
public:
  B (int x) { b = x; }
  int get_b() { return b; }
  };

class C : public A, public B {
  int c;
public:
  C (int x, int y, int z) : A (z), B (y) { c = x; }
  int get_c() { return c; }

void show() {
  cout<<get_a()<<" "<<get_b()<<" "<<get_c()<<endl;
  }
};

int  main() {
  C object(1,2,3);
  object.show();
  return 0;
}
```

Output



Comment

In the above code two parent classes ***A*** and ***B*** were declared where two functions ***get_a(), get_b()*** were declared in each of the classes to get the value of a and b. Then another class ***C*** derived from both **A** and **B** was declared and a constructor was declared in that class to get the access to the private member of the parent classes. After that declaring a class ***C*** type object ***object*** with initialization and calling the ***show()*** function in the main function, the above output was found.