Polymorphism

class car {  
public:
    string color;
    int door;
    void drive() {
        cout << "Driving" << endl;
    }
    virtual void function() {
        cout << "This is car" << endl;
    }
};

class FireEngine : public car {  
public:
    string water;
    void function() {
        cout << "Water!!" << endl;
    }
};

car *c1, c2;
FireEngine fire;
c1 = &fire;
c2 = fire;
c1->function();
c2.function();

Water!!
This is car
Polymorphism

car *c1, c2;
c1->function();

Debug Error!

Program: ...
Module: ...
File:

Run-Time Check Failure #3 - The variable 'c1' is being used without being initialized.

(Press Retry to debug the application)
Polymorphism

car *c1, c2;
c1->function();
Polymorphism

c2 bound statically.  
c1 bound dynamically. (in runtime)

class car {
  public:
    string color;
    int door;
    void drive() {
      cout << "Driving" << endl;
    }
    virtual void function() {
      cout << "This is car" << endl;
    }
};
class FireEngine : public car {
  public:
    string water;
    void function() {
      cout << "Water!!" << endl;
    }
};
In car class, function() method is defined as virtual function. So object c1 will call FireEngine’s function() method.
class Person {
private:
    string name;
    int age;
    string job;
public:
    Person();
    Person(string n, int a, string j);
    void setInformation();
    void printInformation();
};

Person::Person() {
    cout << "Initializing!" << endl;
}

Person::Person(string n, int a, string j) {
    name = n;
    age = a;
    job = j;
}
class car {
private:
    string color;
protected:
    int door;
public:
    void drive() {
        cout << "Driving" << endl;
    }
    virtual void function() {
        cout << "This is car" << endl;
    }
};

class FireEngine : public car{
    string water;
public:
    void function() {
        /* Child class cannot access
           private member of base class */
        color = "Red";
        // protected member of base class
        door = 4;
        // public method member of base class
        drive();
        cout << "Water!!" << endl;
    }
};