```
class Wheel;
class Engine;
class Body;
class Car;

class Visitor {
    public:
    Visitor() { }
    virtual ~Visitor() { }

    virtual void visit(Wheel &) = 0;
    virtual void visit(Engine &) = 0;
    virtual void visit(Body &) = 0;
    virtual void visit(Car &) = 0;
};

class PrintVisitor : public Visitor {
    public:
    PrintVisitor() : Visitor() { }
    virtual ~PrintVisitor() { }

    virtual void visit(Wheel & wheel) {
        std::cout << "Visiting Wheel " << wheel.getSize() <<  std::endl;
    }
    virtual void visit(Engine & engine) {
        std::cout << "Visiting Engine " << engine.getHorsePower() << std::endl;
    }
    virtual void visit(Body & body) {
        std::cout << "Visiting Body" << std::endl;
    }
    virtual void visit(Car & car) {
    std::cout << "Visiting Car" << std::endl;
    }
};

class CarElement {
    public:
    CarElement() { }
    virtual ~CarElement() { }

    virtual void accept(Visitor &) = 0;
};

class Wheel : public CarElement {
    public:
    virtual void accept(Visitor & visitor) {
        visitor.visit(*this);
    }
    int getSize() { return _size; }
    private:
    int _size;
};

class Engine : public CarElement {
```

```
    public:
    virtual void accept(Visitor & visitor) {
        visitor.visit(*this);
    }
    int getHorsePower() { return _hp;}
    private :
    int _hp;
};

class Body : public CarElement {
    public:
    virtual void accept(Visitor & visitor) {
    visitor.visit(*this);

    for (std::vector<Wheel>::iterator i = _wheels.begin(); i != _wheels.end(); ++i) {
        i->accept(visitor);
    }
}

    private:
    std::vector<Wheel> _wheels;
};

class Car : public CarElement {
    public:
    virtual void accept(Visitor & visitor) {
    visitor.visit(*this);

    _engine.accept(visitor);
    _body.accept(visitor);
}

    private:
    Engine _engine;
    Body _body;
};

int main() {
    Car car;
    PrintVisitor visitor;

    car.accept(visitor);
}
```