Lecture 2 Object Oriented Programming

IEEE Spectrum

The Top Programming Languages 2016

What we want:

-robustness

-adaptability

-reusability

C Structure

```
struct Passenger {
    string name;
    int dob[];
    int air_miles;
    struct flights[];
    struct routes[];
};
```

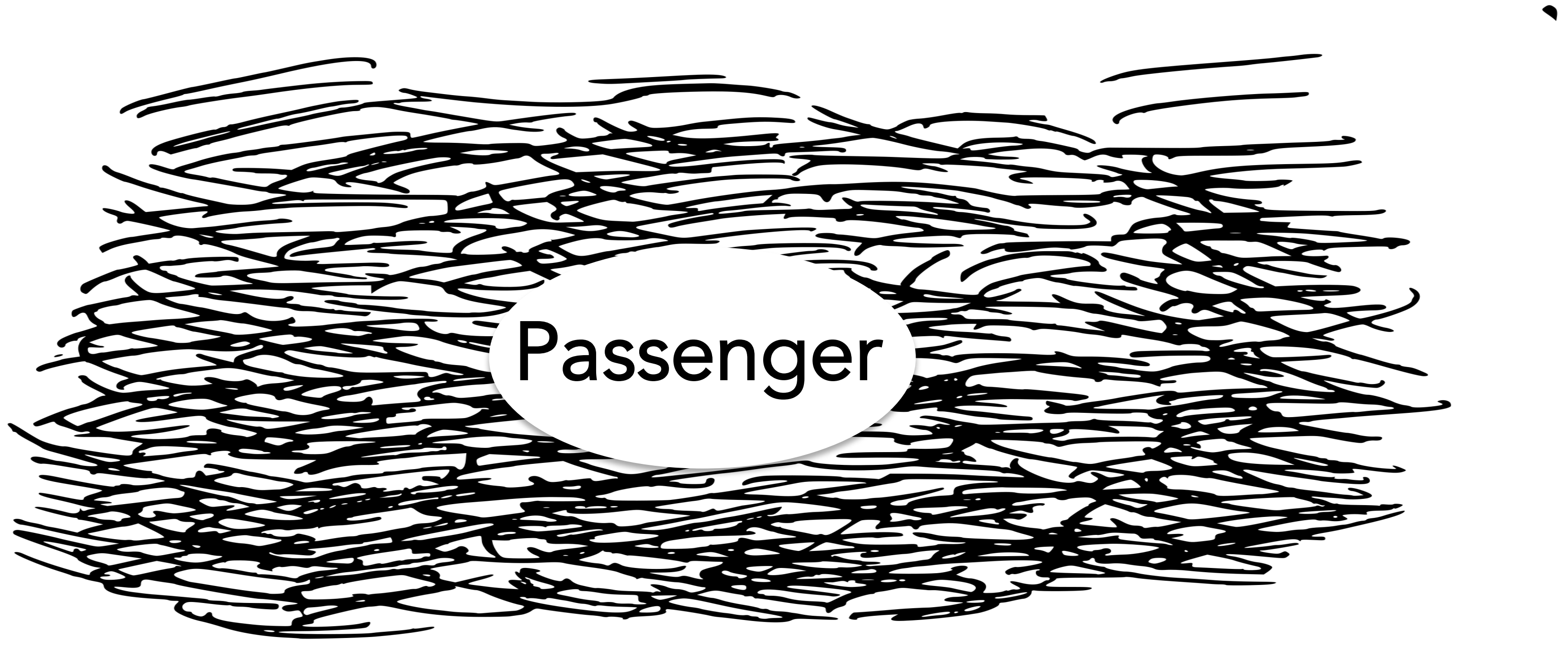
```
struct Passenger Ram;
//Some code
flight L =longestFlight(Ram.flights);

route E =mostEconomic(Ram.routes);
```

Class allows data and functions at one place!

```
class Passenger {
        //data
        …
        //functions
        …
};
```

We want protection from outside code!



private, and public

```
class Passenger {
    private:
        string name;
        int dob[];
        int air_miles;
        flight longestFlight(flight[]);
        route mostEconomic(routes[]);
    public:
        getDetails();
        getName();
        addFlight();
};
```

Abstraction and Encapsulation

Only essential information is provided to the outside world, details are hidden!

```
class Passenger {
    private:
        string name;
        int dob[];
        int air_miles;
        flight longestFlight(flight[]);
        route mostEconomic(routes[]);
    public:
        void getDetails();
        void getName();
        void addFlight();
};
```

Function definition
1. in class
2. outside class

```
void Passenger::getDetails
```

Constructors and Destructor

```
class Passenger {
        //other stuff
        Passenger{string name, int age, int miles}
        //other stuff
        ~Passenger();
};
```

We want easy re-use of the code!

Silver, Gold, and Platinum members!

```
class GoldP: public Passenger{
    private int gold_points;
    …
    …
}
```

Inheritance facilitates code reuse!

Interesting: a Passenger pointer can hold Silver, Gold, and Platinum

Static Binding

Virtual functions

Dynamic Binding

Protected!

Inheritance for specialization or extension!

Encapsulation
Abstraction
Inheritance

Class variables are called objects!

Class Examples

```
class students{
    …
    private int i,j,k;
    public int COMP(i,j){
        //Compare rank
    }
  …
}
```

```
class books{
    …
    private int i,j,k;
    public int COMP(i,j){
        //Compare price
    }
  …
}
```

Object Examples

```
Students st_ref = new Students();

Books bk_ref = new Books();
```

Don’t just get things done, try to find rationale behind anything you do!