Lecture 16: Object Oriented Programming cont.

By Ahmed E. Khaled
Recap to the previous lecture!

• What is an object?
• What is a class?
• How to create class and objects?
• Public vs Private.
Lecture 16 : OOP

• More examples on OOP.
Why do we need Classes / objects?

- The definition of a class is *re-useable* by other object-oriented programs.

- The concept of classes allows a programmer to create any *new data type* that is not already defined in the language itself.
public class person {
    String name;
}

person p1 = new person();
p1.name = "adam";

person p2 = new person();
P2.name = "alice";
Encapsulation

• Hiding internal state and requiring all interaction to be performed through an object's methods.

• This is done through modifier named “private”
public class Car {

    int speed;

}

public static void main(String[] args) {

    Car c1 = new Car();

    c1.speed = 20;

    Car c2 = new Car();

    c2.speed = 0;

}
What is illogical, and how can we fix?

```java
public class car {
    int speed;
}

public static void main(String[] args) {
    car c1 = new car();
    c1.speed = -1;
    car c2 = new car();
    c2.speed = 1000;
}
```
We need to get a control on user can set object’s variables

Step 1 : Encapsulation of the class variables, how?
```java
public class Car {

    int speed;

}

public class Car {

    private int speed;

    void setSpeed (int new_speed){
        speed = new_speed;
    }

    int getSpeed (){ {
        return speed;
    }

}
```
public class car {

    private int speed;

    void set_speed (int new_speed) {
        speed = new_speed;
    }

    int get_speed () {
        return speed;
    }

}

public static void main(String[] args) {

    car c1 = new car();
    c1.speed = -1;

    car c2 = new car();
    c2.speed = 1000;

}
We partially solved it, not completely. Now, no one can access the class members directly. The only way to access the member is through set and get.

```java
public static void main(String[] args) {
    car c1 = new car();
    c1.speed = -1;

    car c2 = new car();
    c2.speed = 1000;
}
```
But still, using set_speed the user can give illogical values!

```java
public static void main(String[] args) {
    car c1 = new car();
    c1.set_speed(-1);
    System.out.println("car 1 speed is : " + c1.get_speed());

    car c2 = new car();
    c2.set_speed(1000);
    System.out.println("car 2 speed is : " + c2.get_speed());
}
```
Now comes the importance of set functions, you can define your own logic values.

```java
public class Car {

    private int speed;

    void set_speed (int new_speed){
        if(new_speed < 0) {
            speed = 0;
        } else if (new_speed > 200) {
            speed = 200;
        } else {
            speed = new_speed;
        }
    }

    int get_speed (){return speed;
}
```
public static void main(String[] args) {

    car c1 = new car();
    c1.set_speed(-1);
    System.out.println("car 1 speed is : " + c1.get_speed());

    car c2 = new car();
    c2.set_speed(1000);
    System.out.println("car 2 speed is : " + c2.get_speed());
}

Create class that models a circle.
Circle has one variable: radius.
Circle has method that returns the area of the circle.
Can we define it this way?

```java
public class circle {
    double radius;

    double calculate_area (){
        double area = radius * radius * 3.14f;
        return area;
    }
}
```
public class circle {

    private double radius;

    void set_radius (double new_radius){
        if(new_radius < 0) {
            radius = 0;
        } else {
            radius = new_radius;
        }
    }

    double get_radius (){ 
        return radius;
    }

    double calculate_area (){ 
        double area = radius * radius * 3.14f;
        return area;
    }

}
public static void main(String[] args) {

    circle c1 = new circle();
    c1.set_radius(4);
    System.out.println("car 1 speed is : " + c1.calculate_area());

    circle c2 = new circle();
    c2.set_radius(-2);
    System.out.println("car 2 speed is : " + c2.calculate_area());
}