Lecture 15: Object Oriented Programming cont.
Recap to the previous lecture!

- What is an object?
- What is a class?
- How to create class and objects?
Lecture 15 : OOP

- Continue OOP.
- Public vs private.
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";
The dot operator?

- `Object_name.class_variable`: means you are accessing the copy of this variable related to this object.

- `Object_name.class_method`: means you are accessing the copy of this method related to this object.
Create Java class in NetBeans

![Image showing a Java project structure in NetBeans]
public class mainclass {

    // Code goes here
}

javax.swing.JFrame frame;
javax.swing.JDialog dialog;
Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name: person

Project: WelcomeJava

Location: Source Packages

Package: cop2800

Created File: nt/NetBeansProjects/ClientServerPair/src/cop2800/person.java
Create Java class
package cop2800;

public class person {

}
package cop2800;

public class person {

    String First_Name;
    int age;

    void display_FullName()
    {
        System.out.println("First name: " + First_Name + " age is: " + age);
    }

}

package cop2800;

public class mainclass {

    public static void main(String[] args) {

        person p1 = new person();
        p1.First_Name = "John";
        p1.age = 30;

        person p2 = new person();
        p2.First_Name = "Alice";
        p2.age = 20;

    }

}
Encapsulation

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

• This is done through modifier named “private”
Why?

- **Public** (non private) member can be accessed from anywhere.

- If something goes wrong with a public member, you need to track large part of the program.

- Accessing private members through methods:
  1. you need to track only one source file (this class java file) in case of errors.
  2. allows you to create all types of checking conditions (boundaries, values, types).
public class person {

    private String name;
    int age;

}

public static void main(String[] args) {

    person p1 = new person();
    p1.name = "John";
    p1.age = 30;

    person p2 = new person();
    p2.name = "Alice";
    p2.age = 20;

}
public class person {

    private String name;
    int age;

    void set_name (String new_name){
        name = new_name;
    }

    String get_name (){    
        return name;
    }

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

    person p1 = new person();
    p1.age = 30;
    p1.set_name("John");
    String person_name = p1.get_name();
    System.out.println("Person name: "+ person_name);

    p1.set_name("Alice");
    System.out.println("Person name: "+ p1.get_name());
}

public class person {

    private String name;
    private int age;

    void set_name (String new_name){
        name = new_name;
    }

    void set_age (int new_age){
        age = new_age;
    }

    String get_name (){{
        return name;
    }

    int get_age (){{
        return age;
    }

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

    person p1 = new person();

    p1.set_name("John");

    p1.set_age(18);

    System.out.println("Person name: "+ p1.get_name());
    System.out.println("Person name: "+ p1.get_age());
}
Try this!

• Create class that models car with speed and direction.
Try this!

• Define setters/getters for the car class variables.
Try this!

• Define method within the class that can update speed with additional new speed.

• Define method within the class that can change the current direction.