Encapsulation

- Wrapping up the implementation of data members and methods in a class.
- this is an application level issue and internal issue.
- it is the process of storing information.

Abstraction

- Hiding unnecessary details and showing valuable information.
- for instance, when someone start a car, does he know all the working of the car ? No.
- this is a design level issue and external issue.
- it is the process of gaining information.

Access

- Here `private` keyword means that the variable or the method can be only accessed in its original class.

Class

```java
package com.inclass.access;

public class A {
    private int num;
    String name;

    // setter
    public void setNum(int num) {
        this.num = num;
    }
    //getter
    public int getNum() {
        return num;
    }
    public A(int num, String name) {
        this.num = num;
        this.name = name;
    }
}
```

Main

```java
package com.inclass.access;

import java.util.ArrayList;

public class Main {
    public static void main (String[] args) {
        A obj = new A (18, "Driptanil");
        System.out.println(obj.num);
    }
}
```

Error

Main

```java
package com.inclass.access;

import java.util.ArrayList;

public class Main {
    public static void main (String[] args) {
        A obj = new A (18, "Driptanil");
        obj.setNum(5);
        System.out.println(obj.getNum());
    }
}
```

Success

- The default access modifier, a variable or method can be only accessed in its original package

Class

```java
package com.inclass.access;

public class B {
    int num;
    String name;

    public B(int num, String name) {
        this.num = num;
        this.name = name;
    }
}
```

Main

```java
package com.inclass.singleton;

import com.inclass.access.B;

public class Main {
    public static void main(String[] args) {
        B object = new B(5, "Drip");
        System.out.println(object.num);
    }
}
```

Error

- By using `public` keyword, the variable or method can be accessed from anywhere.

Class

```java
package com.inclass.access;

public class B {
    public int num;
    String name;

    public B(int num, String name) {
        this.num = num;
        this.name = name;
    }
}
```

Main

```java
package com.inclass.singleton;

import com.inclass.access.B;

public class Main {
    public static void main(String[] args) {
        B object = new B(5, "Drip");
        System.out.println(object.num);
    }
}
```

Success

	public	protected	default	private
Class	+	+	+	+
Package	+	+	+	
Subclass (same pkg)	+	+	+	
Subclass (diff pkg)	+	+		
World	+			

Java contains

1. lang
2. oi
3. util
4. applet
5. awt
6. net

- `hashCode` returns a random integer value formed by using some algorithm.

Code

```java
package com.inclass.access;

public class ObjectDemo {

    int num = 0;


    @Override
    public int hashCode() {
        return super.hashCode();
    }

    public ObjectDemo(int num) {
        this.num = num;
    }

    public static void main (String[] args) {
        ObjectDemo obj = new ObjectDemo(18);
        System.out.println(obj.hashCode());
    }
}
```

Success

2109957412

Code

```java
package com.inclass.access;

public class ObjectDemo {

    int num = 0;
    @Override
    public boolean equals(Object obj) {
        return super.equals(obj);
    }
    public ObjectDemo(int num) {
        this.num = num;
    }
    public static void main (String[] args) {
        ObjectDemo obj = new ObjectDemo(18);
        ObjectDemo obj2 = obj;

        if (obj == obj2) {
            System.out.println("Same Object");
        }
        if (obj.equals(obj2)) {
            System.out.println("Same Object");
        }
        }
}
```

Success

```
Same Object
Same Object
```

Code

```java
package com.inclass.access;

import java.util.Arrays;

public class ObjectDemo {

    int num = 0;

    @Override
    public String toString() {
        return super.toString();
    }
    @Override
    protected void finalize() throws Throwable {
        super.finalize();
    }
    @Override
    public int hashCode() {
        return super.hashCode();
    }
    @Override
    public boolean equals(Object obj) {
        return super.equals(obj);
    }
    public ObjectDemo(int num) {
        this.num = num;
    }
    public static void main (String[] args) {
        ObjectDemo obj = new ObjectDemo(19);
        System.out.println(obj.getClass());
        System.out.println(obj.getClass().getCanonicalName());

System.out.println(Arrays.toString(obj.getClass().getConstructors()));

    }
}
```

Success

```
class com.inclass.access.ObjectDemo
com.inclass.access.ObjectDemo
[public com.inclass.access.ObjectDemo(int)]
```

Code

```java
package com.inclass.access;

import java.util.Arrays;

public class ObjectDemo {

    int num = 0;

    public ObjectDemo(int num) {
        this.num = num;
    }
    public static void main (String[] args) {
        ObjectDemo obj = new ObjectDemo(20);
        System.out.println(obj instanceof ObjectDemo);
        System.out.println(obj instanceof Object);

    }
}
```

Success

```
true
true
```