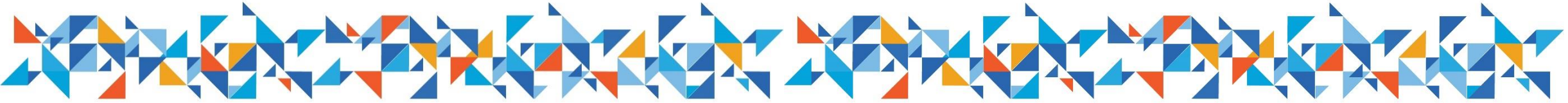
پوهنتون کاردان
KARDAN UNIVERSITY

Object Oriented Programming (Java)

Encapsulation

Learning Outcomes

- Should know about What is Encapsulation?
- Learn about Setters
- Learn about Getters
- Class Practice



Encapsulation

- Encapsulation is a principle of wrapping data (variables) and methods together as a single unit.
- Java encapsulation is referred as **data hiding**.
- But more than data hiding, encapsulation concept is meant for better management or grouping of related data.
- **Points to remember in Encapsulation**
  - **Declare the class variable as private**
  - **Declare the class methods as public**

Encapsulation

- Encapsulation is an OOP technique of wrapping the data and methods.
- In this OOP concept, the variables of a class are always hidden from other classes.
- It can only be accessed using the methods of their current class.

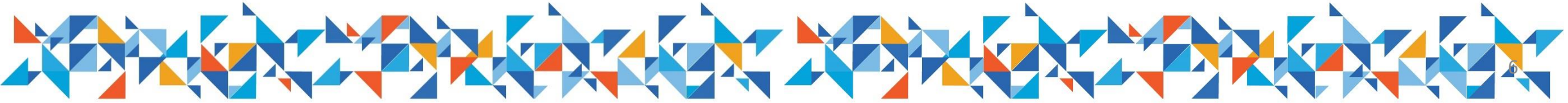
Key Points of Encapsulation in Java:

- **Private Fields:** Encapsulation is implemented by marking class fields (variables) as private, so they cannot be accessed directly from outside the class.

- **Public Getter and Setter Methods:** To access and modify the values of private fields, public methods called getters and setters are provided. This allows control over how variables are accessed and modified.

Benefits of Encapsulation:

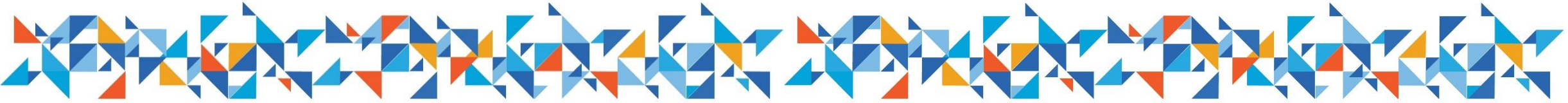
- **Data Hiding**: Restricts access to the internal state of an object.
- **Increased Security**: Protects object integrity by allowing only controlled changes.
- **Improved Maintainability**: Makes it easier to make changes in one part of the code without affecting other parts.
- **Modular Code**: Encapsulated objects can be reused and modified independently.



Setters

- We can get access to the specific code using Getter and Setter methods
- Using setters, we can set and control the values to assign it to class attributes.
- **private** String name;

```
public void setName(String name){
this.name=name
}
```

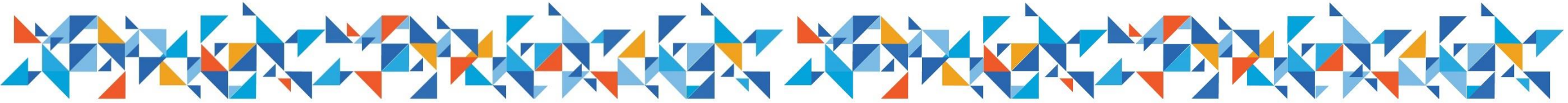


Getters

- Getters are used to grab the value of the class attributes which are in fact, not accessible directly.

- **private** String name;

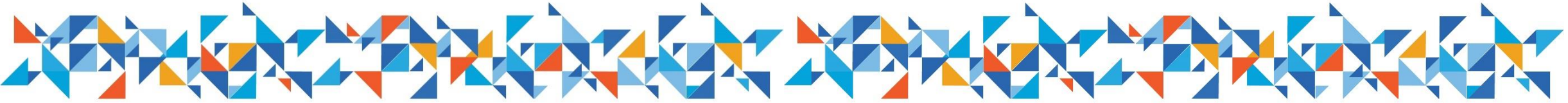
```
public String getName(){
return name;
}
```



Example

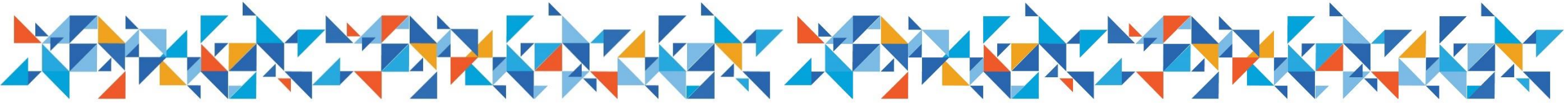
```
public class Student{
//private data member
private String name;
//setter method for name
public void setName(String name){
this.name=name
}
//getter method for name
public String getName(){
return name;
}
}
```

```
class Test{
public static void main(String[] args){
//creating instance of the encapsulated class
Student s=new Student();
//setting value in the name member
s.setName("Ahmad");
//getting value of the name member
System.out.println(s.getName());
}
}
```



Practice

- Consider a Student with the following attributes.
  - Name
  - Father_Name
  - Reg_Number
  - Phone
  - Address.
- Provide the setter and getter methods for each of these attributes?



```java
public class Student {
    private String name;
    private String fname;
    private int regnum;
    private int phone;
    private String address;

    public String getname(){
        return name;
    }
    public void setname(String stdname){
        this.name=stdname;
    }
    public String getfname(){
        return fname;
    }
    public void setfname(String fname){
        this.fname=fname;
                }
        public int getregnum(){
        return regnum;
    }
    public void setregnum(int registration){
        this.regnum=registration;
    }
    public int getmobile(){
        return phone;
    }
    public void setmobile(int phonenumber){
        this.phone=phonenumber;
    }
        public String getaddress(){
        return address;
    }
    public void setaddress(String Address){
        this.address=Address;
    }
}
 class Test{
     public static void main(String[] args) {
         Student obj=new Student();

          obj.setname("Muhammad Zubair");
          System.out.println("Student Name: "+obj.getname());

          obj.setfname("Muhammad Kamran");
          System.out.println("Student's Father Name: "+ obj.getfname());

          obj.setregnum(2022304099);
          System.out.println("Registration #: "+obj.getregnum());

          obj.setmobile(797945679);
          System.out.println("Phone Number is: "+obj.getmobile());

          obj.setaddress("Kardan University Parwan-e-Du Campus Kabul, Afghanistan");
          System.out.println("Address is: "+obj.getaddress());
     }
 }
```

Output

```
run:
Srudent's name: Muhammad Zubair
Student's Father Name: Muhammad Kamran
Registration #: 202304099
Phone number is: 797945679
Address is: Kardan University Parwan-e-Du Campus Kabul,Afghanistan
BUILD SUCCESSFUL (total time: 0 seconds)
```

Consider a bank account having the following attributes:

- Accountnumber
- Name
- Email
- Address
- Phone

Provide the setter and getter methods for each of these attributes. Your output must look like this.

OUTPUT:

```
Welcome to Afghanistan Bank:

Your Account Number is: 98209479297
Your Name is: Mr. Zubair Khan
Your Email is: Zubair1290@gmail.com
Your Address is: Kardan University Parwan –e- Du  square Kabul
Your Phone number is: +93701457892
```

پوهنتون کاردان
KARDAN UNIVERSITY

"Java Program for Bank Account Management Using Encapsulation"

```java
public class BankAccount {
    // Private fields
    private String accountNumber;
    private double balance;

    // Constructor
    public BankAccount(String accountNumber, double initialBalance) {
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }
    // Public method to deposit money
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        } }

    // Public method to withdraw money
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid withdrawal amount.");
        }}
    // Public method to get the current balance
    public double getBalance() {
        return balance;
}}
```

```java
public class BankAccountTest {
    public static void main(String[] args) {
        // Creating a new bank account with an initial balance
        BankAccount account = new BankAccount("12345", 1000.0);

        // Testing deposit
        account.deposit(500.0);  // Depositing 500
        System.out.println("Current Balance: " + account.getBalance());

        // Testing withdraw
        account.withdraw(200.0);  // Withdrawing 200
        System.out.println("Current Balance: " + account.getBalance());

        // Testing invalid withdrawal
        account.withdraw(2000.0); // Attempt to withdraw more than the balance
        System.out.println("Current Balance: " + account.getBalance());
    }
}
```

Output

```
run:
Deposited: 500.0
Current Balance: 1500.0
Withdrew: 200.0
Current Balance: 1300.0
Invalid withdrawal amount.
Current Balance: 1300.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

پوهنتون کاردان
KARDAN UNIVERSITY

Thank You…!