ENCAPSULATION ASSIGNMENT

**Q1. What is Encapsulation in java ? Why is it called data hiding?  
Ans.** Binding of data and corresponding method in a single unit together known as encapsulation. Uses of private keyword makes the data hide which means data can be accessible within the same class not from other class.  
To prevent data from outside the world (class) make it protected and restricted for anyone use .

Encapsulation = Data hiding + Data abstraction

**Q2. What are the important features of Encapsulation?  
Ans.** The features of encapsulation are :-

* Data would not be accessible from other class.
* Data would be hide.
* Security of our programme would be increase.
* The main thing use of encapsulation helps us to understand our code if any bug comes.
* We can provide control access , security increases.
* Method would be hide as per abstraction.

**Q3. What are Getter and Setter method in java . Explain with an example?  
Ans. Setter:-** Setter method is used to set the value of a instance variable in a class.  
**Syntax :-**

* Compulsory the method name should be start with set keyword.
* It should be public.
* Return type should be void.
* Compulsory it should have some argument.

**Getter:-** Getter method is used foe get the value from a instance in a class.

**Syntax:-**

* Compulsory the method name should be start with get keyword.
* It should be public.
* Return type should not be void
* Compulsorily it should not have any argument.
* package Encapsulation;  
   class student{  
   private int age;  
   private String name;  
   public void setAge(int age){  
   this.age = age;  
   }  
   public int getAge(){  
   return age;  
   }  
   public void setName(String name){  
   this.name = name;  
   }  
   public String getName(){  
   return name;  
   }  
   public void print(){  
   System.*out*.println(age);  
   System.*out*.println(name);  
   }  
   }  
  public class Practice1 {  
   public static void main(String[] args) {  
   student obj = new student();  
   obj.setAge(24);  
   obj.setName("Ayush");  
   obj.getAge();  
   obj.getName();  
   obj.print();  
   }  
  }

**Q4. What is the us of this keyword in java explain with an example?  
Ans.** If we declare the name of instance variable and local variables same inside method then name clash happen between both of them during compilation, because when our code is compile jvm will consider both variables as local variables, but in reality they both have different , one is instance and other is local. This problem is known as Shadowing problem for eliminating this problem we used ‘this’ keyword.  
**this** keyword is indicating or refer as same object which is instance and both variable consider as originals.

class student{  
 private int age;  
 private String name;  
 public void setAge(int age){  
 this.age = age;  
 }  
 public int getAge(){  
 return age;  
 }  
 public void setName(String name){  
 this.name = name;  
 }  
 public String getName(){  
 return name;  
 }  
 public void print(){  
 System.*out*.println(age);  
 System.*out*.println(name);  
 }  
 }  
public class Practice1 {  
 public static void main(String[] args) {  
 student obj = new student();  
 obj.setAge(24);  
 obj.setName("Ayush");  
 obj.getAge();  
 obj.getName();  
 obj.print();  
 }  
}

**Q5. What is the advantages of java encapsulation?  
Ans.** The advantages of java are:-

* A class can have complete control over its data members and data methods.
* The class will maintain its data members and methods as read-only.
* Data hiding prevents the user from the complex implementations in the code.

**Q6. How to achieve encapsulation in java . Give an example?  
Ans.** By just following simple steps:-

* Declaring the variables as private in a class.
* Providing getter and setter method to modify and views the variables values.
* class student{  
   private int age;  
   private String name;  
   public void setAge(int age){  
   this.age = age;  
   }  
   public int getAge(){  
   return age;  
   }  
   public void setName(String name){  
   this.name = name;  
   }  
   public String getName(){  
   return name;  
   }  
   public void print(){  
   System.*out*.println(age);  
   System.*out*.println(name);  
   }  
   }  
  public class Practice1 {  
   public static void main(String[] args) {  
   student obj = new student();  
   obj.setAge(24);  
   obj.setName("Ayush");  
   obj.getAge();  
   obj.getName();  
   obj.print();  
   }  
  }