**Encapsulation -Assignment**

Question 1: -**What is encapsulation in java ?Why is it called data hiding?**

Ans: -Encapsulation in Java refers to integrating data (variables) and code (methods) into a single unit. In encapsulation, a class's variables are hidden from other classes and can only be accessed by the methods of the class in which they are found.It is important to declare this class as private.

If we talk about data encapsulation so, **Data encapsulation** hides the private methods and class data parts, whereas **Data hiding** only hides class data components.

Syntax:

<Access\_Modifier> class <Class\_Name> {

private <Data\_Members>;

private <Data\_Methods>;

}

Question 2: -**What are the important features of Encapsulation ?**

Ans: -Combine the data of our application and its manipulation at one place.

Encapsulation Allow the state of an object to be accessed and modified through behaviors.

Reduce the coupling of modules and increase the cohesion inside them.

Question 3: -**What are the getter and setter methods in java Explanation with an example?**

Ans: -In Java, getter and setter are two conventional methods that are used for retrieving and updating value of a variable.

**EX:-**

**package hello;**

**class data\_hide {**

**private int age;**

**private String name;**

**public int getAge() {**

**return age;**

**}**

**public void setAge(int age) {**

**this.age = age;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**}**

**public class getter\_setter {**

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

**data\_hide dh=new data\_hide();**

**dh.setAge(20);**

**dh.setName("vikash chauhan");**

**System.*out*.println(dh.getAge());**

**System.*out*.println(dh.getName());**

**}**

**}**

Question 4: -**What is the use of this keyword explain with an example?.**

Ans: -There can be a lot of usage of java this keyword.

**1.**this can be used to refer to the current class instance variable.

**2.**this can be passed as an argument in the method call and constructor call.

**3.**this can be used to invoked current class method implicitly.

**4.**this can be used to invoked current class constructor.

**5.**this can be used to return the current class instance from the method.

**Example: -**

**package hello;**

**class data\_hide {**

**private int age;**

**private String name;**

**public data\_hide(int age,String name) {**

**this.age=age;**

**this.name=name;**

**}**

**public int getAge() {**

**return age;**

**}**

**public void setAge(int age) {**

**this.age = age;**

**}**

**public String getName() {**

**this.getAge();**

**return name+":"+age;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**}**

**public class getter\_setter {**

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

**data\_hide dh=new data\_hide(12,"vivek chauhan");**

**//dh.setAge(20);**

**//dh.setName("vikash chauhan");**

**//System.out.println(dh.getAge());**

**System.*out*.println(dh.getName());**

**}**

**}**

Question 6: -**What is the advantage of encapsulation?.**

Ans: -**Data Protection:** The program runner will not be able to identify or see which methods are present in the code. Therefore we don't get any chance to change any specific variable or data and hinder the running of the program.

**Flexibility:** The code which is encapsulated looks more cleaner and flexible, and can be changed as per the needs. We can change the code read-only or write-only by getter and setter methods. This also helps in debugging the code if needed.

**Reusability:** The methods can be changed and the code is reusable

**Testing code is easy:** Encapsulated code is easy to test for unit testing.

Question 6: -**How to achieve encapsulation in java ?Give an example.**

Ans:-Encapsulation can be achieved by Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

**package** hello;

**class** data\_hide {

**private** **int** age;

**private** String name;

**public** data\_hide(**int** age,String name) {

**this**.age=age;

**this**.name=name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**public** **class** getter\_setter {

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

data\_hide dh=**new** data\_hide(12,"vivek chauhan");

dh.setAge(20);

dh.setName("vikash chauhan");

System.out.println(dh.getAge());

System.***out***.println(dh.getName());

}

}