

Encapsulation Assignment

1. What is Encapsulation in java? Why is it called Data hiding?

Ans: Binding of data and corresponding methods into a single unit is called Encapsulation.

In Encapsulation data is hidden by using private method hence it is called as Data hiding.

2. What are the important features of Encapsulation?

Ans: important features of Encapsulation:

1. Security
2. Enhancement becomes easy
3. Maintainability and modularization becomes easy
4. flexibility

3. What are getter and setter method in java explain with example?

Ans: setter and getter methods are used to set and get the value to and for the object variable.

```
Eg. class Student1 {
    private int age;
    public void setAge(int age) {
        this.age=age;
    }
    public int getAge() {
        return age;
    }
}
public class SetGet {
    public static void main(String[] args) {
        Student s = new Student();
        s.setAge(45);
        System.out.println(s.getAge());
    }
}
```

4. What is the use of this keyword explain with an example?

Ans: this keyword can be used to refer current class instance variable.

```
Eg. class Student1 {
    private int age;
    public void setAge(int age) {
        this.age=age;
    }
    public int getAge() {
        return age;
    }
}
public class SetGet {
    public static void main(String[] args) {
        Student s = new Student();
        s.setAge(45);
        System.out.println(s.getAge());
    }
}
```

5. What is the advantage of encapsulation?

Ans: Advantages of Encapsulation:

1. Security
2. Enhancement becomes easy
3. Maintainability and modularization becomes easy
4. flexibility to the user to use the system very easily.

6. How to achieve Encapsulation in java? Give example?

Ans: We can create a fully encapsulated class in Java by making all the data members of the class private.

```
Eg. class Student1 {
    private int age;
    public void setAge(int age) {
        this.age=age;
    }
    public int getAge() {
        return age;
    }
}

public class SetGet {
    public static void main(String[] args) {
        Student s = new Student();
        s.setAge(45);
        System.out.println(s.getAge());
    }
}
```

Constructor Assignment

1. What is a constructor?

Ans: whenever we are creating an object some piece of code will be executed automatically to perform initialization of an object. This piece of code is nothing but a constructor.

2. What is Constructor chaining?

Ans: Constructor chaining is the process of calling a constructor inside a constructor. We can do it in two ways by using this() keyword for chaining constructors in the same class or by using super() keyword for chaining constructors from the parent class.

3. Can we call a subclass constructor from a superclass constructor?

Ans: Yes, A subclass constructor can be called from a superclass constructor by using super() keyword.

4. What happens if you keep a return type for a constructor?

Ans: If we keep a return type for a constructor, then it will become a method of the class.

5. What is No-arg Constructor?

Ans: A constructor which not accept any argument is called as No-arg Constructor.

6. How is No-argument constructor different from the default constructor?

Ans: The default constructor is created by java compiler if we do not create a constructor but, if we create constructor with 0 arguments then it will be No-argument constructor.

7. When do we need constructor overloading?

Ans: if we want to initialize different variables at the different occasion then we need a constructor overloading with Different parameters.

8. What is default constructor explain with example?

Ans: if we don't create any constructor inside a class then java compiler will automatically create a default constructor which will be invoked at the time of object creation.

```
class Demo{
    //public disp()
    // {
    //     // default constructor
    // }
    public void disp()
    {
        System.out.println("Hello,World");
    }
}

public class ConstructorEx {
    public static void main(String[] args) {
        Demo d = new Demo();
        d.disp();
    }
}
```

Static Keyword Assignment

1. Why do we need a static keyword in java explain with an example?

Ans: if we want to execute some initialization and method to execute directly without creating an object for that , we need a static keyword.

```
Eg. public class StaticEc {
    static int a;
    static int b;
    static
    {
        System.out.println("static block");
        a=10;
        b=10;
    }
    public static void main(String[] args) {
        System.out.println("main method");
    }
}
```

2. What is class loading and how does the java program actually executes?

Ans: class loading is the process of loading class files into the JVM at runtime. It is responsible for loading classes from various sources and making them available to the JVM for execution. First class is loaded by class loader after that the file goes into the runtime data area where various memory areas are available which stores the methods , variables etc. after that the file is interpreted by interpreter and JIT compiler then the file is executed by OS.

3. Can we mark a local variable as static?

Ans: NO, java does not allow static local variable.

4. Why is the static block executed before the main method in java?

Ans: static block is executed first because the static block is executed at time of class loading.

5. Why is static method also called as class method?

Ans: A static method is a method that belongs to a class rather than an instance of a class. Hence static method also called as class method.

6. What is the use of static block in java?

Ans: normally a static block is used to perform initialization of the static variables.

7. Difference between static and instance variables?

Ans: *Static variables:*

1. these variables will get memory in the method area.
2. if the value does not change from object to object then we need to use static variables.
3. inside a static area we can access static variables only.
4. static variables are created using static keyword.

Instance variable:

1. these variables will get memory in heap area.
2. if the value is changes form object to object then we need to use “non-static” i.e instance variables.

8. Difference between static and non static members?

Ans:

1. Static variables can be accessed using class name Non static variables can be accessed using instance of a class
2. Static variables can be accessed by static and non static methods. Non static variables cannot be access inside a static method.
3. Static variables reduce the amount of memory used by a program. Non static variables do not reduce the amount of memory used by a program
4. Static variables are shared among all instances of a class. Non static variables are specific to that instance of a class.
5. Static variable is like a global variable and is available to all methods. Non static variable is like a local variable and they can be accessed through only instance of a class.

Practical Assignment Questions:

1. Create a class that keeps track of the number of instances created. Implement a static variable and method to accomplish this.

```
Ans: public class InstanceCount {
    static int count=0;
    public InstanceCount(){
        count++;
    }
    public static void main(String[] args) {
        InstanceCount ic = new InstanceCount();
        InstanceCount ic2 = new InstanceCount();
        InstanceCount ic3 = new InstanceCount();
        System.out.println(ic.count);
    }
}
```

2. Write a program and create a constructor with parameters and initialize the variable using a constructor.

```
Ans: class Cons {
    int a;
    int b;
    public Cons(String name)
    {
        a=10;
        b=20;
        System.out.println(name);
    }
}
public class ConsEx{
    public static void main(String[] args) {
        Cons c = new Cons("rohan");
        System.out.println(c.a + c.b);
    }
}
```

3. Use a private keyword for a variable and use setter and getter methods to initialize and print the values.

```
Ans: class Student1 {
    private int age;
    public void setAge(int age)
    {
        this.age=age;
    }
    public int getAge()
    {
        return age;
    }
}
public class SetGet {
    public static void main(String[] args) {
        Student1 s = new Student1();
        s.setAge(45);
        System.out.println(s.getAge());
    }
}
```

4. Write a program to call a method without creating an object of a class

```
Ans: class Emp{
    static int salary;
    static int sal()
    {
        salary=30000;
        return salary;
    }
}

public class AutoMethodCall {
    public static void main(String[] args) {
        System.out.println(Emp.sal());
    }
}
```

5. Write a program which has static block and constructor overloading, initialize variables using constructors and print it.

```
Ans: class Demo4{
    private String name;
    private int age;
    private byte roll;
    static {
        System.out.println("static block");
    }
    public Demo4(String name, int age){
        this.name=name;
        this.age=age;
    }
    public Demo4(byte roll){
        this.roll=roll;
    }
    public void print()
    {
        System.out.println("name and age is: "+name+" "+ age);
        System.out.println("roll is: "+roll);
    }
}

public class StaticAndCons {
    public static void main(String[] args) {
        Demo4 d = new Demo4("rohan",18);
        Demo4 d1 = new Demo4((byte) 5);
        d1.print();
    }
}
```