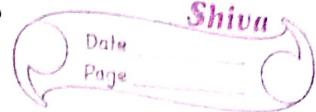


Abhinav Raj Sri Varshava.
Java 2.0 Hindi English.



Assignment - 0. (constructor)

Q-1 what is encapsulation?

- (a) In Java, constructor is basically a special way of calling of a method while creating an object of a class.
- (b) The constructor method have it's name of it's class's name and it has no return type not even void.

Example - public Student {

 private name;

 private int age;

 public Student (name, age) {

 this.name = name;

 this.age = age;

}

}

public static void main (String [] args) {

 Student S₁ = new Student (Ram, 21);

 Student S₂ = new Student (Raju, 20);

}

Q-2 What is constructor using - ?

Constructor using refers to the process of one constructor calling another constructor within the same class or in its Super class.

Q-3 Can we call as Sub class constructor from a Super class constructor?

- No, it is not possible to call Sub class constructor from a Super Class constructor in Java.

Ex- class Student {

 name

 private String Name;

 public Student () {

Q-4 What happen if you keep a return type from a constructor?

- In Java, constructor do not have a return type so if we keep some return type with it, we will get an compilation error.

Ex- constructor Student in class test.

error : Student cannot be applied

to the given type

Q-5 What is No-argument constructor?

② A no argument constructor is basically a constructor without any argument from the calling source.

③ Do not get any information from the calling point of the class.

Q-6

have no-argument constructor? is different from default constructor?

- ② a no-fault constructor is automatically generated by the Java compiler if no explicit constructor are defined in a class and we can't use initialization value using no-fault constructor in the class.

- ③ A no-argument constructor is explicitly defined in the code and can be used to initialize an object's state.

When no argument are provided during object creation.

Q-7 When do we need constructor over loading?

- ① Constructor overloading is a method of creating multiple constructor of different parameter of a single class name:

use need constructor overloading when

we are given more than one object of same class having different parameters.

Q-8 What is a default constructor?

In Java when we create an object of a class and do not create any constructor explicitly then behind the scenes Java compiler create a default constructor of that object inside the class of object having zero receiving parameters as well as an empty body.

Example- class Student {

private String name;

* Default → public Student () {
constructor; }

II Empty body

public static void main(String[] args) {
Student s = new Student();
}

Assignment - of (encapsulation)

Q-1

What is encapsulation in Java?

Why it is called Data hiding?

*1

Encapsulation in Java Encapsulation
means to making Data hiding and
Data protection of as clean from

the outside clean
the directly.

Q-2

Security increase - Encapsulation make
our software more secure and tractable
as the Data access directly from
outside the class is not possible.

Q-3

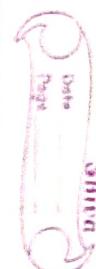
What are the getter and Setter method in
Java Explain?

* Data hiding - Using Encapsulation method
we are able to make our class
in size a class private, so that no outside
the class, no method or way can
access the Data or manipulate directly.

Q-4

Getter () - Gets method is used to
get some Data or Information in order
to initialize the Data with the parial

Setter () - Setter method one use
to set value of the instance variable
through the parameter if receive
from method call.



Q-4 Use of 'this' keyword give some example.

This keyword - In Java keyword 'this' is used at the time of Construction and Initialization of some Data to the instance variables of any particular object.

Example - class Student {

```
private String name;
private int age;
```

```
public Student (name, age) {
    this.name = name;
    this.age = age;
}
```

```
public static void main (String [] args) {
    Student s1 = new Student ("Raju", 20);
```

```
Student s2 = new Student ("Raju", 21);
```

```
public Example of
public static void main (String [] args) {
```

```
fruits f1 = new fruits("apple", 100);
```

Q-5 Advantage of encapsulation ?

- * Data Privacy.
- * Data hiding.
- * Data abstraction.
- * Access control.
- * Code reusability.
- * Security.
- * Evolution of compatibility.

Ques to achieve Encapsulation in Java ?
Give an Example ?

```
public class fruit {
    private String name;
    private int price;
```

```
public fruit (name, price) {
    this.name = name;
    this.price = price;
```

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

```
fruits f1 = new fruits("apple", 100);
```

Assignment-9: (practice questions).

Q-1:

create a class that keeps track of the number of instance created. Implement a static variable and method to accomplish this.

```
public static Student {
```

```
    static String name;
    static int age;
```

```
public static Result () {
```

```
    int math = 90;
    int Bio = 100;
```

```
    S.o.p ( math );
    S.o.p ( Bio );
```

```
    this.age = age;
```

```
public Student (String name, int age) {
```

```
    this.name = name;
```

```
System.out.println ("Your name is " + name + " and
```

```
you are " + age + " years old");
```

```
public clean Demo {
```

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

```
Student s1 = new Student (Ram, 20);
```

```
Student s2 = new Student (Sham, 10);
```

```
Student . age = 20;
```

```
} // output
```

Your name is Ram and you are 20 years old.
Your name is Sham and you are 10 years old.

Q-2: write a program and a constructor with parameter and initialize the variable using the constructor.

```
public Student {
```

```
private String name;
private int age;
```

Date _____
Page _____
Shiva

Date _____
Page _____
Shiva

Q-9 use a **private** keyword for a variable and use **getter** and **getter** method to initialize and print the value.

```
public class Student {
```

```
    private String name;  
    private int age;
```

```
    public void setName (String name, int age) {
```

```
        this.name = name;  
        this.age = age;
```

```
    }  
    public void getName () {
```

```
        System.out.println ("Name " + name +  
                           " Age " + age);
```

```
    }  
}
```

```
public class Main {
```

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

```
        Student s = new Student ("Ram", 21);  
        System.out.println (s.getName());
```

Output:

Hello Ram it is a constructor call.

```
public class Main {
```

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

```
        Student s = new Student ("Abinav", 21);
```

```
        System.out.println (s.getName());
```

Q-10 write a program to use an method without creating an object of a class.

```
public class Main {
```

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

```
        Student s = new Student ("Ram", 21);
```

```
        System.out.println (s.getName());
```

}

```
public class Student {
```

```
    public String name;
```

```
    public void getName () {
```

```
        System.out.println (name);
```

}

Q-5 write a program which has static block and constructor overloading, initialise variable using constructor and print it.

part-1

```
public class Student {
```

```
    * private String name;
```

```
* } . * }
```

```
System.out.println("This is a  
static block");
```

// output

Ram

Ram 21.

```
* } . * }  
void Student (String name) {  
    this.name = name;  
    System.out.println(name);  
}
```

```
* } . * }
```

```
void student (String name, int age) {
```

```
System.out.println("Name : " + name);  
System.out.print("Age : ");
```

```
public class Demo {  
    public static void main (String [] args) {  
        Student s1 = new Student ("Ram",  
                                  20);  
    }  
}
```

Assignment - 10 (Static).

Q-1

Why do we need Static Keyword in Java?

Q-2

In Java, the 'static' keyword is used to define class-level members that are shared among all instances of class.

It can be applied to variable, method, and nested classes.

Q-3 Variable, methods or blocks created using

Static keyword, are loaded onto the memory when the class of it loaded on the memory.

Q-4

loading — the JVM's class loader collects and read the byte code of a class or interface.

It is created once and till the end of the program execution it will be provide the memory.

Q-5

The class is also known as informal representation of the class called a "class" object.

Linking — linking is divided into three

Sub-categories : verification, preparation, and resolution.

1. Static Variables.

2. Static Method

4. Static Block

Class loading and how does the Java program actually executes?

Class loading in Java is a process by which the JVM loads the classes and interfaces into memory during the execution of a Java program.

The class loading process consists of three steps —

② Initialization - this is the final step, the JVM executes the static initialization code of the class.

- Static variables are assigned their explicit initial values, and any static initializer blocks are executed in the order they appear in the source code.

Q-3 Can we mark a local variable as static?

- No, it is not possible to mark a local variable as 'static' in Java.
- The 'static' keyword is used to define class-level members, such as variables and methods, that are shared among all the instances.
- Local variables are variables declared within a method, constructor or block, and they are scoped to that specific block.
- They are created and destroyed each time the block is executed.

Q-4: Why static block executed before the main method in Java?

- When a Java program starts, the JVM first loads the class into the memory and initialize its static members, including static variables and static blocks.
- The 'main' method then executed as the entry point of the program.

Q-5: Why is a static method is also called as a class method?

A Static Method in Java is also called a class method because it is associated with the class itself rather than with instance of the class.

Q-6: uses of static block in Java?

- for class level Initialization.
- for utilization of memory.
- for saving memory.
- Exception handling during class initialization
- Static configuration or setup.

Q-1

Static Variable

Instance Variable

- (a) they are created once and shared among all objects of the class. they are created each time of creation of any object.
- (b) they're created inside the heap memory. they are also created inside a heap memory.
- (c) the life of a static variable is from starting of the program till end of the execution of program. the life of a IV are start with its object creation and end with object destruction.

Q-2

Static member

non-Static member

- (a) It belongs to a class itself rather than any certain instance of class. they are associated with individual instances of the class.

(b)

ClassName::Static method(); Object::nonStatic method();

(c)

Static members are allocated memory only once when the class loaded into memory. they are allocated memory for each time any instance of the class.