package package1;

// Java program to illustrate If statement

class IfDemo {

public static void main(String args[])

{

Sample2333 SampleObj23 = new Sample2333();

// System.out.println(SampleObj23.add() );

// System.out.println(SampleObj23.add(1,2) );

System.out.println(SampleObj23.add("Ramana SAI") );

}

}

class Sample2333

{

int a=90;

byte b= 89;

String name = "Rachana";

//non praprmeter

int add(){

//comment

String name = "sdfkgsdf";

int c= 23+23;

System.out.println("I am a non paramtereted and with rtetuurn value function");

return c ;

}

int add(int a, int b){

//comment

int c= a+b;

System.out.println("I am parametered nd with rtetuurn value function");

return c;

}

boolean add(String str){

//comment

System.out.println("Hellow " +str +"can you hear me ?");

return false;

}

}

1. **why the output is showing false?**

Here the return statement is false so that the out put is showing false.If the return statement is true the output will be true.

1. **Why do we only have string arguments?**

Because everything you pass as argument to your program on the command line can be represented as a string. Not as doubles or integers.

1. **What is the difference between c and java access specifiers?**

C does not have access modifiers because C is not an object oriented language. Access modifiers (or access specifiers) are keywords in object-oriented languages that set the accessibility of classes, methods, and other members

1. **what is difference b/w constructor and function in the class**

* Constructor is used to initialize an object whereas method is used to exhibits functionality of an object.
* Constructors are invoked implicitly whereas methods are invoked explicitly.
* Constructor does not return any value where the method may/may not return a value.
* In case constructor is not present, a default constructor is provided by java compiler. In the case of a method, no default method is provided.
* Constructor should be of the same name as that of class. Method name should not be of the same name as that of class.

1. **who gives default constructor? what are other responsibilities of jvm.**

In case constructor is not present, a default constructor is provided by java compiler. The JVM has two primary functions: to allow Java programs to run on any device or operating system (known as the "Write once, run anywhere" principle), and to manage and optimize program memory.

* Loads code
* Verifies code
* Executes code
* Provides runtime environment

1. **Is java call by reference or call by value**

There is only call by value in java, not call by reference. If we call a method passing a value, it is known as call by value. The changes being done in the called method, is not affected in the calling method.

1. **who is providing the extra functions is java?**
2. **what happens if final keyword is applied on variable, class and method?**

Java final keyword is a non-access specifier that is used to restrict a class, variable, and method. If we initialize a variable with the final keyword, then we cannot modify its value.

If we declare a method as final, then it cannot be overridden by any subclasses. And, if we declare a class as final, we restrict the other classes to inherit or extend it.In other words, the final classes can not be inherited by other classes.

1. **why string is passed in main method?**

Because by passing String arrays , we can pass all the necessary parameters like options/arguments related to the program in the form of String easily. There can be several parameters! Also, all the other datatypes can be easily converted from String!

1. **Does JVM provide a default constructor?**

In case, programmer does not provide any constructor in class definition – JVM provides a default constructor to the class in runtime. In default constructor, name of the constructor MUST match the class name and it should not have any parameters.

1. **Who provides extra functions in main method?**
2. **Difference b/w java 8 and java7**

Java SE 7 was the first major release of the programming language under Oracle’s ownership and stewardship since it acquired Sun Microsystems in 2010. Java 7 was a significant upgrade to the Java model which accommodates some major upgrades to the programming language including language enhancements, multiple exceptions handling, JVM support for dynamically-typed languages, and more. Java SE 8 is a revolutionary release of the programming language which took the model to a whole new level.  Wit Java 8, the programming language bring along its anticipated feature called the Lambda Expressions which changed the entire coding paradigm for the Java platform.  
  
Java 8 brings its own new specialized API for Date and Time manipulation.

1. **when and why collections were added?**

The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes ([ArrayList](https://www.javatpoint.com/java-arraylist), Vector, [LinkedList](https://www.javatpoint.com/java-linkedlist), [PriorityQueue](https://www.javatpoint.com/java-priorityqueue), HashSet, LinkedHashSet, TreeSet).

1. **Difference b/w collection and collections.**

**Collection:** Collection is a [interface](https://www.geeksforgeeks.org/interfaces-in-java/) present in java.util.package. It is used to represent a group of individual objects as a single unit. It is similar to the container in the [C++](https://www.geeksforgeeks.org/c-plus-plus/) language. The collection is considered as the root interface of the collection framework. It provides several classes and interfaces to represent a group of individual objects as a single unit.

The [List](https://www.geeksforgeeks.org/list-interface-java-examples/), [Set](https://www.geeksforgeeks.org/set-in-java/), and [Queue](https://www.geeksforgeeks.org/queue-interface-java/) are the main sub-interfaces of the collection interface. The map interface is also part of the java collection framework, but it doesn’t inherit the collection of the interface. The add(), remove(), clear(), size(), and contains() are the important methods of the Collection interface.

**Collections:** Collections is a utility class present in java.util.package. It defines several utility methods like sorting and searching which is used to operate on collection. It has all static methods. These methods provide much-needed convenience to developers, allowing them to effectively work with [Collection Framework](https://www.geeksforgeeks.org/collections-in-java-2/). For example, It has a method *sort()* to sort the collection elements according to default sorting order, and it has a method *min()*, and *max()* to find the minimum and maximum value respectively in the collection elements.