IQ :

Why main method of java accepts String as a parameter ?

* String is a collection of alpha-numeric character so it can hold all different types of values.
  Our main method accept String array as a parameter so we have two advantages :

   a) We can accept all **different** types of values.
   b) Due to String array we can hold **multiple** values

System.out.println() ;

                          System.out.println();

              ┌──────────────────┼──────────────────┐
    System is a   out is a reference   It is predefined method
    predefined class   variable of type   available in
    available in   PrintStream,   java.io.PrintStream class
    java.lang   PrintStream is a
    package   predefined class
                             available in java.io
                             package.

* It is an output statement in java through which we can print the data on the console.
  In the PrintStream class we have multiple **println() and print()** methods are available
  with different parameter.

* Actually System.out.println() represents HAS-A relation as shown below.

    ┌──┐
    │ public final class System   │
    │ {   │
    │   public static final java.io.PrintStream out = null; //HAS-A │   .
    │ }   │
    └──┘

         Note : System class **HAS-A** PrintStream class
───
//WAP in java two add two numbers
public class Addition
{
    public static void main(String[] args)
    {
        int x = 100;
        int y = 200;
        int z = x + y;
        System.out.println(z); //300
    }
}
───
How to provide user friendly message to the Client

Case 1 :

//Providing user friendly message in 'C' language style :
   public class Addition
   {
       public static void main(String[] args)
       {
           int x = 100;
           int y = 200;
           int z = x + y;
           System.out.**printf**("Sum is :%d",z); //C language style
           System.out.println();
       }
   }

Case 2 :

//Java Style
Behavior of '+' operator in Java :   Operands
-----------------------------------   /\
                                               a + b;
                                                  ↓
                                               Operator

* While working with '+' operator, If any of the operand is **String type then the '+'
  Operator will behave as a String Concatenation Operator.**

   1 + 1 = 2   [Here '+' operator works as Arithmetic Operator]

   "1" + 1 = 11 [Here '+' operator works as String concatenation Operator]

   22 + "44" = 2244 [Here '+' operator works as String concatenation Operator]

   public class Addition
   {
       public static void main(String[] args)
       {
           int x = 100;
           int y = 200;
           int z = x + y;
           System.out.println("Addition is :"+z);
       }
   }
───
//Addition of two number without using 3rd variable
public class Sum
{
    public static void main(String[] args)
    {
        int a = 100;
        int b = 200;
        System.out.println("Sum is :"+a+b); //100200
        System.out.println(+a+b);//300
        System.out.println(""+a+b);//100200
        System.out.println("Sum is :"+(a+b));
    }
}
───
public class IQ
{
    public static void main(String[] args)
    {
        String s1 = 40 + 40 + "NIT" + 20 + 20; //80NIT2020

        System.out.println(s1);   .
    }
}
───
Command Line Argument :

              public void accept(int x) //Parameter
              {
              }

              public static void main(String [] args) //args is also parameter
              {
              }

              accept(10); //Argument
                          ──────→ Here If we pass any argument then It is called
              main( ); ;   **Command Line Argument**

* If we pass any argument to the main method then It is called **Command Line Argument.**

* By using Command Line Argument, We can pass some value **at runtime.**

* The advantage of command line Argument is "Single time compilation and number of times
  execution".

                   public class Command
                   {
                       public static void main(String[] args)
                       {
                           System.out.println(args[0]);
                       }
                   }

                   javac Command.java (Compilation)
                   java Command Java James
                                  [0]   [1]
//Programs

Case 1 :
----
public class Command
{
    public static void main(String[] args)
    {
        System.out.println(args[0]);
    }
}

javac Command.java
java Command Java James
Output : Java

Case 2 :
----
public class Command
{
    public static void main(String[] args)
    {
        System.out.println(args[1]);
    }
}

Note : We need to re-compile the code for new .class file

javac Command.java
java Command Java James
Output is : James

Case 3 :
----
public class Command
{
    public static void main(String[] args)
    {
        System.out.println(args[0]);
    }
}

javac Command.java
java Command   [Not Passing any value at runtime]
Exception is Generated : java.lang.ArrayIndexOutOfBoundsException

Case 4 :

WAP to print First Name and Last Name by using commnd Line Argument :
class FullNameByUsingCommand
{
    public static void main(String[] args)
    {
        System.out.println(args[0]);
    }
}

javac FullNameByUsingCommand .java
java FullNameByUsingCommand "James Gosling"
Output is : James Gosling