

Continue



DSA to Development: A Complete GuideBeginner to AdvanceJava Backend Development - LiveIntermediate and AdvanceTech Interview 101 - From DSA to System Design for Working ProfessionalsBeginner to AdvanceFull Stack Development with React & Node JS - LiveBeginner to AdvanceJava Programming Online Course [Complete Beginner to Advanced]Beginner to AdvanceC++ Programming Course Online - Complete Beginner to AdvancedBeginner to AdvancePage 2Our website uses cookiesWe use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy PolicyGlobal variables are the variables that can be accessed anywhere in the java program. Java being a strictly object-oriented programming language doesn't have the concept of a global variable. But by using static and final keywords, a variable can have all the functionality of a global variable.Global variables are the variables that can be accessed from anywhere in the program. These are different from local variables that we define in methods or main method which can only be used in that particular method. The scope of local variables is limited to that method only, i.e. they cannot be accessed outside the method they are declared in. Whereas, global variables can be used by any of the methods in a program.In programming languages like C, C++, Python, etc. global variables can be declared anywhere in the program using the global keyword. But Java's object-oriented programming structure makes it difficult to reference variables in multiple places. Thus, unlike other languages, Java officially doesn't have the concept of global variables.However, we can achieve the functionality of global variables using existing properties of Java, such as static and final keywords.Java is purely an object-oriented programming language and thus, everything in Java is wrapped in a class. All the related variables and methods are kept together, making the program structured.Global variables are to be declared outside the methods which are against OOP principles. In Object-Oriented Programming Language, everything is an object and all the programs and data reside within classes and objects. Global variables also have other security threats, and thus in java, we don't have global variables.Let us take an example of global variable ver as in the code below:Now one of the team members interprets it as the vertical distance from the center and the other interprets it as a version number.To avoid such confusing situations, we use variables inside classes and functions. If the variable is related to the version then it must be stored in the about class as follows:If it is related to the vertical distance then it is stored in Grid class.The scope of a variable refers to the part of the program where we can access the variable. In its scope, we can edit, update, print, or copy the value of a variable. Global variables are the variables whose scope is the entire program. In other words, these variables are accessed from anywhere in the program.Since Java does not have the concept of global variables, we can simulate their functionality by creating constants using static or static final keywords, referencing a class, or implementing an interface, similar to how other programming languages implement global variables.Let us see each of the ways in detail to create global variables in Java.Create Global Variables by Using the Reference Class in JavaReference class is a class where we store all the global values together if they are used commonly in all the parts of the program. If multiple global variables are required, then we use this method of defining global variables.A reference class is a class that stores all the global variables and they can be accessed simply by Reference.className anywhere in the program.Let us take an example of a Circle class that uses the reference variable PI to calculate the area of the circle.Output:In the above example, a class called Reference doesn't have any other functionalities, only constants are declared. Each constant is public thus, anyone can access it. They all are final which implies their value cannot be changed. Also, we don't require the object to call the variable as it is static.The public class Circle has a method called area that takes the radius of a circle and returns the area. It uses the reference variable PI to calculate the area.In the main method, we use the area method to calculate the area of a circle of radius 23.Create Global Variables by Using Interfaces in JavaWe can define all the global variables in an interface and the class that requires to access the variables can implement this interface. All the variables in the interface are public, static and final by default.Let us take an example of class Gravitation that has two methods. The first one calculates the force exerted by one body on another, and the second one calculates the weight using mass of a body.Output:We have defined an Interface Constant that holds the constant values such as the Universal Gravitation constant, the mass of earth, the mass of the moon, etc.Gravitation class implements this interface. The method Force calculates the force exerted by Earth on Moon. The Weight method calculates the weight of the body by its mass and acceleration due to gravity. The simplest way to create global variables is using the static keyword and declaring the variable outside the method in the class. As we are using the static keyword, we can directly call the variable. If we don't use static keywords, the object of the class is required to access the variable.Let us take an example of the Greet Class and Message Class to greet everyone.Output:In the example above, the variable firstMessage is a global variable. It can be accessed by both the classes Greet and Message as both the classes belong to the same program and firstMessage is a public(global) and static variable. As it is static, we do not require an object of the Greet class to use the variable in the Message class. We can call it using className.variableName as in the above example Greet.firstMessage. In this way, we can achieve the functionality of the global variable in Java.Create Global variables by using static & final keywords in JavaWe can also declare static variables as final, i.e. they cannot be changed later in any part of the program once declared and initialized. It is mostly used to declare constants in Java.Note:Non-static variables can be changed inside methods.Let us take an example of Calculate class that calculates the amount to be paid using MRP(Maximum Retail Price).Output:In the example above the deliveryCharge is declared as final and static which means it cannot be changed in the program for any value of MRP i.e. it is fixed. We calculate the amount to be paid using the TotalPrice method and return it to the user. The variables that are declared within the method are known as the local variables. They can be accessed only inside the method they are declared in. We need to pass these as a parameter to other functions if we wish to use them outside our function. Global variables need not be returned or passed, they are available to all the functions.Major differences between Local and Global variables: Local VariablesGlobal VariablesThese are declared inside a function and their scope is limited to that function only.Global variables are declared outside the function and are accessible throughout the program.Local variables get created when the function or method in Java is called and is lost when the function is terminated.It is created when the global execution of the program starts and is lost when the program terminates.Data sharing is not possible.Data Sharing is possible throughout the program with the global variables in java.When the local variable is modified, its value does not affect any other part of the program.When global variables are manipulated then it affects all the other parts of the program where they are used.If we want to access the value of a local variable outside the method we need to pass it as a parameter to another method.We are not required to pass it as a parameter as it is available to all the methods of the program.It is stored on the stack.It is stored at a fixed location in the memory.Global variables in Java can be accessed by all the methods in the program.We need to declare them only once outside the methods.It can be accessed multiple times from any of the methods in the program.For example, we can declare PI outside the methods as it is a constant value and needs to be used again and again.If so many variables are declared global, then they stay in the memory from the beginning till the end and the program may run out of the memory.Any function or method can change the value of the global variable if it is not declared as final as it is accessible to all the methods of the program. Furthur this changed value can affect the calculations done by other functions.Global variables can be accessed by all the methods in the program, whereas local variables can only be used in the method they are declared in.As Java is Object Oriented Programming Language, it doesn't have global variables.We can use static variables in a class, static final variables in a class, create a new class to store static variables only, or create an interface to store variables to give the functionality of global variables in Java.CsharpProgrammingServer Side Programming A local variable is used where the scope of the variable is within the method in which it is declared. They can be used only by statements that are inside that function or block of code. Live Demousing System;public class Program { public static void Main() { int a = 100; // local variable Console.WriteLine("Value: "+a); } }OutputValue:100Global VariablesC# do not support global variables directly and the scope resolution operator used in C++ for global variables is related to namespaces. It is called global namespace alias.If you have a type that share an identifier in different namespace, then to identify them use the scope resolution operator. For example, to reference System.Console class, use the global namespace alias with the scope resolution operator global::System.ConsoleExampleusing myAlias = System.Collections.namespace Program { class Demo { static void Main() { myAlias::Hashtable hTable = new myAlias::Hashtable(); hTable.Add("A", "1"); hTable.Add("B", "2"); hTable.Add("C", "3"); foreach (string str in h.Keys) { global::System.Console.WriteLine(str + " " + h[str]); } } } } Get certified by completing the course Get Started Prerequisite: Variables in CIn a programming language, each variable has a particular scope attached to them. The scope is either local or global. This article will go through global variables, their advantages, and their properties.The Declaration of a global variable is very similar to that of a local variable. The only difference is that the global variable is declared outside any function. We can take an example by assuming that we have a chair at our house and one in our school/college then we can say that the chair at our home can only be accessed by the people living inside the home but the chair in our college can be used by any student or faculty.Example: C // C program to show declaration of global variable#include int x = 5; // global variableint main() { int y = 10; // local variable return 0;}Global variables do not stay limited to a specific function, which means that one can use any given function to access and modify the global variables. The initialization of these variables occurs automatically to 0 during the time of declaration. Also, we generally write the global variables before the main() function.Use of the Global VariableThe global variables get defined outside any function- usually at the very beginning/top of a program. After this, the variables hold their actual values throughout the lifetime of that program, and one can access them inside any function that gets defined for that program.As already stated earlier, any function can access a global variable. It means that once you execute a program, its global variable will be available for use throughout the running of the entire program.Advantages of Global VariableGlobal variables can be accessed by all the functions present in the program.Only a one-time declaration is required.Global variables are very useful if all the functions are accessing the same data.Disadvantages of Global VariableThe value of a global variable can be changed accidentally as it can be used by any function in the program.If we use a large number of global variables, then there is a high chance of error generation in the program.Example: C // C program to update global variables#include int a, b; // global variables void add() { // we are adding values of global a and b i.e. 10+15 printf("%d", a + b);} int main(){ // we are now updating the values of global variables // as you can see we dont need to redeclare a and b // again a = 10; b = 15; add(); return 0;} In lesson 7.3 – Local variables, we covered that local variables are variables defined inside a function body. Local variables have block scope (are only visible within the block they are declared in), and have automatic duration (they are created at the point of definition and destroyed when the block is exited).In C++, variables can also be declared outside of a function. Such variables are called global variables.Declaring global variablesBy convention, global variables are declared at the top of a file, below the includes, in the global namespace. Heres an example of a global variable being defined:#include // Variables declared outside of a function are global variablesint g x {}; // global variable g_xvoid doSomething(){ // global variables can be seen and used everywhere in the file g_x = 3; std::cout