Java error missing return statement

Having some difficulties in figuring out where my return statement is in the exercise of the First Directive. This is keeping me from moving on. Returns are something my brain is still struggling to wrap my head around. Any advice on how to correct my mistake would be greatly appreciated. Here is the error code that the compiler is giving me: PrimeDirective.java:31: error: return declaration missing } ^ 1 error // Import declaration: import java.util.ArrayList; PrimeDirective class { // Add your methods here: Boolean audience isPrime (number int) { if (number == 2) { return true; } more if (number < 2) { false return; } to (int i = 2; i < number; i++) { if (number % i == 0) { false return; } } } } main public static void (String[] args) { PrimeDirective pd = new PrimeDirective(); numbers int[] = {6, 29, 28, 33, 11, 100, 101, 43, 89}; System.out.println(pd.isPrime(6)); } } 1 How I reformed your brackets a little to clarify the intention. The real question is, does the compiler know that its method is guaranteed to return a value? If this does not happen, you need to make it explicitly return a value (in this case, Boolean) as a default case at the end (in case the control flow drops). boolean audience isPrime (int number) { if (number == 2) { true return; } more if (number < 2) { false return; } to (int i = 2; i < number; i++) { if (number % i == 0) { false return; } } } 2 Likes Thanks for the reply and organizing my brackets more cleanly! I don't know what statement I need to complete with a return value that hasn't been completed yet? the first if the declaration is completed with a true return and return false, should I complete the 2nd with another statement and return true;? for (int i = 2; i < number; i++) { if (number % i == 0) { false return; } I'm not sure what you mean by default or how to produce the standard case at the end. Do I need to format it as a switch statement? Maybe I need to go through the tutorial if again to solidify the concept. Or can I use a break statement to tell the program to exit for each loop? public static boolean isPrime (int number) { if (number == 2) { true return; } more if (number < 2) { false return; } to (int i = 2; i < number; i++) { if (number % i == 0) { false return; } } } public static booleano isPrime (number int) { if (number == 2) { true return; } more if (number < 2) { false return; } to (int i = 2; i < number; i++) { if (number = i == 0) { false return; } Hi, sorry if the terminology was confusing, I didn't want to use a switch declaration – but I meant a lot like a switch statement to provide the method with something to do in case all the alternatives are false. How do you do it? You. You don't need a pause after the return (as the return already breaks the loop and something else { true return; } It may be a possible way not to throw an error. Although you need to consider the logic behind using it and its placement. I would consider getting it wrong on the false return side at the end of the method and then refactoring it if it's not needed so deep. Okay, so I corrected the statement if-else to return true in the right place; but then I wasn't printing anything until I added another false return statement as you suggested. boolean audience isPrime (int number) { if (number == 2) { true return; } more if (number < 2) { false return; } to (int i = 2; i < number; i++) { if (number = = = 0) { false return; } So this false return statement on the isPrime method supports effectively terminates the Boolean method? I'm confused why my program would need this false return claim if the .isPrime() method is already using the first if (number % i == 0) to execute the false return code at number 6. Is it just a fault device that the compiler requires not to go into an infinite loop? We can go back with assumptions. Each method must return what its definition promises. This is a high-priority check to compile. I think anything that throws ambiguity about whether the promised return is existing, throws an error. So unless there is a control flow with a guaranteed statement again that returns a value, the compiler would have to test or break its own rule on returns to validate the method. That's what I think is happening. It's been a while since I've played real java. Sign in or register to answer this question. The missing return statement on the java error is mainly faced by beginners. You can familiarize yourself with these types of common errors by coding simple java projects. Let's understand the mistake first. This error occurs at compile time. The root cause of the error as the name suggests when the return declaration is missing in the program. Read Also: Fix illegal expression start error in public class Java HelloWorld { Public static string printingHelloWorld() { System.out.println(HelloWorld of JavaHungry!); } main public static void (String args[]) { printHelloWorld(); } } I'm using the Java HelloWorld program as an example. If you compile the above code using the HelloWorld javac command below.java you will have the missing return declaration error. There are two ways to fix it. 1.1 In the printHelloWorld() method we declare the return type as String. But if you notice that there is no return declaration within the printHelloWorld() method. Just add this as shown below. public static string printingHelloWorld() { System.out.println(HelloWorld from JavaHungry!); Return Alive is Awesome; } Now compile and run your using commands below Javac HelloWorld.java java HelloWorld Yahoo!! Your Yours is settled. HelloWorld from JavaHungry! 1.2 The above situation can also be resolved by changing the return type to empty without providing any return declaration. public static void printingHelloWorld() { System.out.println (HelloWorld from JavaHungry!); } Now build and run your program using below helloworld javac commands.java java HelloWorld Yahoo!! your problem is solved. HelloWorld from JavaHungry! If you declare the return declaration within itself/while/to but not at the end of the method that contains them, then you will have missing return declaration error as shown below. /** * Java program to demonstrate * Missing return declaration in java error * inside if /while/for * @author Subham Mittal */ HelloWorld public class { public static string printingHelloWorld() { int i = 0; if (i == 0) { return within the block if; } to (i=0; i < 9 ;i++) { return in to loop; } while (i < 9) { return in while loop; } //Return declaration missing to method } main public static void (String args[]) {Hello printWorld(); } } } If you compile the above code, then you will have missed return declaration error. Fix it by adding the return statement to the method similar to what we did in case 1 above, that is, adding Live line return is Amazing; Note: Assuming that the return method type is not null. You must provide the return declaration for the method, which should be the last statement in the method. There is a special case, which happens when you provide return declarations within the se/else block, as shown below. What's the way out? HelloWorld public class { public static string printingHelloWorld() { int i = 0; if (i == 0) { return within the block if; } something else { return within the block of another party; } main public static void (String args[]) { printHelloWorld(); System.out.println (HelloWorld from JavaHungry!); } } The above code will compile well and you will have the following output: HelloWorld from JavaHungry! That's all for today. Please mention in the comments if you have any questions related to the return declaration lost in the java error. There are many types of errors that can be encountered during java software development, from, but most are preventable. We've gathered 50 of the most common Java software errors, complete with code examples and tutorials to help you work around common coding issues. For more tips and tricks for coding better Java programs, download our comprehensive Java Developer Guide, which is packed with everything you need to grow your Java game – from tools to the best websites and blogs, YouTube channels, Twitter influencers, LinkedIn groups, podcasts, must-see events, and more. If you are with .NET, you should also check our guide to the 50 most common .NET software errors and how to avoid them. But if your current challenges are related to Java, read on to learn about the most common and their alternative solutions. Compiler error messages are created when Java software code runs through the compiler. It is important to remember that a compiler can throw many error messages for an error. Then correct the first error and rebuild. This could solve many problems.1. ... expected This error occurs when something is missing in the code. Often this is created by a missing semicolon or closing parentheses.private static double volume (String solid, double height, double areaBase, double radius) { double vol; se (solidom.equalsIgnoreCase(){ vol=(4.0/3)*Math.pi*Math.pow(ray,3); }) { vol=Math.pi*Math.pow(raiom,2)*alturam; } else {vol=(1.0/3)*Math.pi*Math.pow(raiom,2)*alturam; } } return vol; } Often this error message does not identify the exact location of the problem. To find it:Make sure that all opening parentheses have a corresponding closing parenthesis. See in the line before the indicated Java line of code. This Java software error is not noticed by the compiler until further in the code. Sometimes a character as an opening parenthesis should not be in Java code in the first place. Thus, the developer did not make a closing parenthesis to balance the parentheses. Check out an example of how a lost parenthesis can create an error (@StackOverflow).2. Closed sequence literalThe unclosed sequence literal error message is created when the literal sequence ends without quotation marks, and the message appears on the same line as the error. (@DreamInCode) A literal is a source code of a value. abstract class public NFLPlayersReference { private static Runningback[] nflplayersreference; static private players quarterback[] ; static private players WideReceiver[] nflplayers; public static void main (String args[]){ Runningback r = new Runningback(Thomlinsion); Quarterback q = new quarterback (Tom Brady); WideReceiver w = new WideReceiver (Steve Smith); NFLPlayersReference[] NFLPlayersReference; Run();// { NFLPlayersReference = new NFLPlayersReference [3]; nflplayersreference[0] = r; players[1] = q; nflplayers[2] = w; for (int i = 0; i < nflplayersreference.length; i++) { System.out.println(My name is + nflplayersreference[i].getName()); nflplayersreference[i].run(); nflplayersreference[i].run(); nflplayersreference[i].run(); System.out.println (NFL offensive threats have great execution skills!); } empty private static Run() { System.out.println (Not yet implemented); } } Commonly, this happens when: The literal sequence does not end with quote marks. This is easy to fix by closing the literal sequence with the required citation tag. The literal string goes beyond a line. Long string literals can be divided into several literals and concatenated with a plus sign (+). The brands quote that are part of the literal string do not escape with a rear rear bar a discussion of the non-closed Java software literal error message. (@Quora) 3. Illegal start of an expressionThere are numerous reasons why an illegal start error of an expression occurs. It turns out to be one of the least useful error messages. Some developers say it's caused by bad code. Expressions are usually created to produce a new value or assign a value to a variable. The compiler expects to find an expression and cannot find it because the syntax does not meet expectations. (@StackOverflow) It is in these statements that the error can be found.} ADD IT HERE public void newShape(String shape) { switch (shape) { case Line: Line of shape = new line (startX, startY, endX, endY); shapes.add(line); break; oval case: Shape oval = new Oval (startX, startX, endX, endY); shapes.add(oval); break; case Rectangle: Rectangle of shape = new Rectangle (startX, startY, endX, endY); shapes.add(); rectangle break; default: System.out.println(ERROR. Check the logic.); } } } } REMOVE IT FROM HERE } Browse discussions on how to resolve an expression's illegal start error. (@StackOverflow) 4. Can't find symbolThis is a very common problem because all identifiers in Java need to be declared before they are used. When the code is being compiled, the compiler does not understand what the handle means. There are many reasons why you may receive the message cannot find symbol: The spelling of the identifier when declared may not be the same as when it is used in code. The variable was never declared. The variable is not being used in the same scope in which it was declared. The class wasn't imported. Read a full discussion about the error that cannot find symbol and code examples that create this problem. (@StackOverflow) 5. Public class XXX must be in the fileThe public class message XXX must be in the file occurs when the class XXX file name and the Java program name do not match. The code will only compile when the class and java file are the same. (@coderanch)javaapplicationtion3; public class robot { int xlocation; int ylocation; Name of the string; int ccount sattics = 0; public robot (int xxlocation, int yylocation, String nname) { xlocation = xxlocation; ylocation = yylocation; name = nname; ccount++; } } public class JavaApplication1 { main public static void (String[] args) { robot firstRobot = new Robot(34,51,yossi); System.out.println(numebr of robots is now + Robot.ccount); } } To fix this problem: Name the class and archive it. Make sure that the case of both names is consistent. Here's an example of the Public Class XXX error that should be in the file. (@StackOverflow) 6. Incompatible typesIncompatible types is an error in logic that occurs when an assignment statement tries to a variable with a type expression. It often comes when the code tries to put a text string in an integer - or vice vice This is not a Java syntax error. (@StackOverflow)test.java:78: error: incompatible types return stringCompilder.toString(); ^ required: int found: String error 1 There is not really an easy fix when the compiler gives a message incompatible types: There are functions that can convert types. The developer may need to change what the code should do. Here's an example of how to try to assign a sequence to an integer created the incompatible types. (@StackOverflow) 7. Invalid method declaration; required return type This Java software error message means that the return type of a method was not explicitly indicated in the signature.public class Circle method { private double radius; public circler (double r) { radius = r; } public diameter() { double d = radius * 2; return d; } There are a few ways to trigger the invalid method declaration; return type error required: Forgetting to declare the typeS the method does not return a value, then empty needs to be indicated as the type in the method signature. The names of the builders don't have to be the state type. But if there is an error in the constructor name, then the compiler treats the constructor as a method without a indicated type. Follow an example of how the constructor naming triggered the invalid method declaration; type of return required. (@StackOverflow) 8. The <X>method in the class cannot be applied to <Y>certain typesThis Java software error message is one of the most useful error messages. Explains how the method signature is calling the wrong parameters. RandomNumbers.java:9: error: Method generate Numbers in Random Numbers class cannot be applied to certain types; generate Numbers(); required: int[] found:generateNumbers(); reason: Lists of actual and formal arguments differ in length The called method is to expect certain arguments defined in the method declaration. Check the method declaration and call carefully to make sure they are compatible. This discussion illustrates how a Java software error message identifies the incompatibility created by arguments in the method declaration and method call. (@StackOverflow) 9. Missing return declarationThe missing return message occurs when a method does not have a return declaration. Each method that returns a value (a non-empty type) must have a statement that literally returns that value so that it can be called outside the OpenFile() method.public String[] launches

The map map<String, double=> { new HashMap(); FileReader fr = new FileReader (money.txt); BufferedReader br = new BufferedReader(fr); try{ while (br.ready(){String str = br.readLine(); String[] list = str.split( ); System.out.println (list); } capture } (IOException e){ System.err.println(Error - IOException!); } } There are a few reasons why a compiler launches the missing return declaration message: A return statement was simply omitted by </String,> </Y> </X> </X> method did not return any value, but the empty type was not declared in the method signature. Here's an example of how to fix the missing Java software claim error. (@StackOverflow)10. Possible loss of precisionPossible loss of precision occurs when more information is assigned to a variable than it can contain. If that happens, the pieces will be thrown away. If this is fine, then the code needs to explicitly declare the variable as a new type. A possible loss of precision error usually occurs when:Attempting to assign a real number to a variable with an entire data type. Trying to assign a double to a variable with an entire data type. This explanation of the Primitive Data Types in Java shows how the data is characterized. (@Oracle)11. reached the end of the file while parsingThis error message usually occurs in Java when the program is missing the closing curly strap (}). Sometimes it can be quickly corrected by placing it at the end of code.public class mod_MyMod extends BaseMod public String Version() { return 1.2_02; } public void AddRecipes(CraftingManager recipes) { recipes.addRecipe (new ItemStack (Item.diamond), new Object[] { #, Character.valueOf('#'), Block.dirt} } The above code results in the following error:java:11: reached the end of the file while analyzing } Coding utilities and proper code encoding can make it easier to find these unbalanced keys. This example shows how missing devices can create the error message that you reached when parsing. (@StackOverflow)12. Unreachable declarationUnreachable declaration occurs when a claim is written in a place that prevents it from being executed. Usually this is after a break or return declaration.para(;;) { break; ... // unreachable statement } int i=1; if(i==1) ... something else ... // dead code Often simply move the return statement will correct the error. Read the discussion on how to fix unreachable instructional Java software error. (@StackOverflow)13. variable <X>may not have been initializedThat occurs when a local variable declared within a method has not been initialized. It can occur when a variable without an initial value is part of a statement.int x; if (condition) { x = 5; } System.out.println(x); x may not have been initialized Read this thread on how to avoid triggering the variable error <X>may not have been initialized. (@reddit)14. Operator.. Object This often happens when the Java code tries to use a type string in a calculation. To fix it, the string needs to be converted to an integer or float. Read this example of how non-numeric types were causing a Java software warning that an operator cannot be applied to a type. (@StackOverflow)15. cannot= be= applied= to= java.lang.object,java.lang.object= this= often= happens= when= the= java= code= tries= to= use= a= type= string= in= a= calculation.= to= fix= it,= the= string= needs= to= be= converted= to= an= integer= or= float.read= this= example= of= how= non-numeric= types= were= causing= a= java= software= error= warning= that= an= operator= cannot= be= applied= to= a= type.= (@stackoverflow)15.=></ cannot be applied to java.lang.Object,java.lang.Object This often happens when the Java code tries to use a type string in a calculation. To fix it, the string needs to be converted to an integer or float. Read this example of how non-numeric types were causing a Java software error warning that an operator cannot be applied to a type. (@StackOverflow)15. > são usados para tipos que não estão em sua definição.operador</X> </X> </X> </X> TypesThe error types unconvertable occurs when java code tries to perform an illegal conversion. TypeInvocationConversionTest.java:12: unconvertable types found : java.util.ArrayList <> <? extends= typeinvocationconversiontest.interface1=?> > needed: java.util.ArrayList <> <?? > > lessRestrictiveClassList = (ArrayList <> <?? > >) classList; ^ For example, Booleans cannot be converted into an integer. Read this discussion about finding ways to convert unseeable types into Java software. (@StackOverflow)16. Missing return valueYou will receive the missing return value message when the return claim includes an incorrect type. For example, the following code:public class savingsAcc2 { private double balance; private double interest; public savingsAcc2() { balance = 0.0; interest = 6.17; } Public savingsAcc2 (double initbalance, double interested) { balance = initBalance; interest = interested; } Acc2 Deposit Public Savings (double amount) { balance = balance + amount; return; } Public savingsAcc2 withdraw (double value) { balance = balance = value; return; } Public savingsAcc2 addInterte (double interest) { balance = balance = balance * (interest / 100) + balance; } double public getBalance() { return balance; } Returns the following error:SavingsAcc2.java:29: return of the lost return value; ^ SavingsAcc2.java:35: return lost return; ^ SavingsAcc2.java:41: return lost return; ^ 3 errors There is usually a return statement that returns nothing. Read this discussion on how to avoid the java missing value software error message. (@coderanch)17. cannot return a value of the method whose result type is emptyThis Java error occurs when an empty method tries to return any value, as in the following example:public static void movement() { System.out.println(What do you want to do?); Scanner scan = new Scanner (System.in); int userMove = scan.nextInt(); return userMove; } public static void usersMove (String playerName, int gesture) { int userMove = move(); if (userMove == -1) { break; } This is often corrected by changing to method signature to match the type in the return statement. In this case, void cases can be changed to int:public static int move() { System.out.println(What do you want to do?); Scanner scan = new Scanner (System.in); int userMove = scan.nextInt(); return userMove; } Read this discussion on how to correct the error cannot return a value of the method whose result type is empty. (@StackOverflow)18. variable not static... cannot be referenced from a static context This error occurs when the compiler tries to access nonstatic variables from a static method (@javinpaul):p ublic class StaticTest { int count main public static void (String args[]) throws IOException { count++; //compiler error: the nonstatic variable count cannot be to be from a static context } } To correct the non-static variable . . . . cannot be referenced from a static context error, two things can be done:The variable can be declared static in the signature. Code can create an instance of a nonstatic object in the static method. Read this tutorial that explains the difference between static and non-static variables. (@sitesbay)19. non-static method... cannot be referenced from a static context This problem occurs when Java code tries to call a nonstatic method in a nonstatic class. For example, the following code:class sample { private int age; empty public setAge(int a) { age=a; } public int getAge() { return age; } public static void main (String args[]) { System.out.println(Age is:+ getAge()); } } } Would return this error:Exception on the main line java.lang.Error: Unresolved build problem: Cannot make a static reference to the nonstatic getAge() method of type Sample To call a nonstatic method from a static method is to declare an instance of the class calling the nonstatic method. Read this explanation of what the difference is between nonstatic methods and static methods.20. (array) <X>uninitializedYou will receive the message (array) <X>not initialized when an array has been declared but not started. The arrays are fixed in length, so that each array needs to be initialized to the desired length. The following code is acceptable:AClass[] array = {object1, object2} As is:AClass[] array = new AClass[2]; ... array[0] = object1; array[1] = object2; But no:AClass[] array; ... array = {object1, object2}; Read this discussion on how to boot arrays in Java software. (@StackOverflow) Runtime exceptions21. ArrayIndexOutOfBoundsExceptionThis is a runtime error message that occurs when code tries to access an array index that is not within the values. The following code would trigger this exception:String[] name = {tone, dick, harry}; for(int i = <=name.length; i++)= {= system.out.print(name[i]= +");= }= here's= another= example= (@dukeu):int[]= list=new int[5];= list[5]=33; illegal= index,= maximum= index= is= 4= array= index= start= at= zero= and= end= at= one= less= than= the= length= the= array.= often= it= is= fixed= by= using=></=name.length;> < instead= of=></> <= when defining the limits of the array index. Check out this example of how an index triggered the ArrayIndexOutOfBoundsException Java software error message. (@StackOverflow)22. StringIndexOutOfBoundsExceptionThis is an issue that occurs when the code attempts to access part of the string that is not within the bounds of the string. Usually this happens when the code tries to create a substring of a string that is as longer as the parameters are set at. Here's an example (@javacodegeeks):public class StringCharAtExample { public static void main(String[] args) { String str = Java Code when= defining= the= limits= of= the= array= index.check= out= this= example= of= how= an= index= triggered= the= "arrayindexoutofboundsexception"= java= software= error= message.= (@stackoverflow)22. = "stringindexoutofboundsexception"this= is= an= issue that= occurs= when= the= code= attempts= to= access= part= of= the= string= that= is= not= within= the= bounds= of= the= string.= usually= this= happens= when= the= code= tries= to= create= a= substring= of= a= string= that= is= not= as= longer= as= the= parameters= are= set= at.= here's= an= example= (@javacodegeeks):public= class= stringcharatexample= {= public= static= void= main(string[]= args)= {= string= str=></=" when defining the limits of the array index. Check out this example of how an index triggered the "ArrayIndexOutOfBoundsException" Java software error message. (@StackOverflow)22. "StringIndexOutOfBoundsException"This is an issue that occurs when the code attempts to access part of the string that is not within the bounds of the string. Usually this happens when the code tries to create a substring of a string that is not as longer as the parameters are set at. Here's an example (@javacodegeeks):public class StringCharAtExample { public static void main(String[] args) { String str = Java Code > 0; i</X> </X> </X> System.out.println(Length: + str.length()); The following declaration throws an exception, because //the request index is invalid. char ch = str.charAt(50); } Like array indexes, string indexes start at zero. When indexing a string, the last character is at one less than the length of the string. The Java StringIndexOutOfBoundsException software error message usually means that the index is trying to access characters that are not there. Here is an example that illustrates how StringIndexOutOfBoundsException can occur and be fixed. (@StackOverflow)23. NullPointerExceptionA NullPointerException will occur when the program tries to use an object reference that does not have a value assigned to it (@geeksforgeeks).// A Java program to demonstrate that it invokes a method // about null causes NullPointerException import java.io.*; GFG class { main public static void (String[] args) { // Initializing String variable with null value String ptr = null; // Check if ptr.equals is null or works well. try { // This line of code launches NullPointerException // because ptr is null if (ptr.equals(gfg)) System.out.print(Same); else System.out.print(Not Same); } catch(NullPointerException e) { System.out.print(NullPointerException Caught); } } The Java program often raises an exception when: A statement references an object with a null value. Attempt to access a class that is defined but is not assigned a reference. Here's the discussion about when developers can find NullPointerException and how to handle it. (@StackOverflow)24. NoClassDefFoundErrorNoClassDefFoundError will occur when the interpreter does not find the file containing a class with the main method. Here is an example of DZone (@DZone):If you compile this program:class A { // some code } public class B { public static void main (String[] args) { A a = new A(); } Two files .class are generated: A.class and B.class. By removing file A.class and running file B.class, you will receive the NoClassDefFoundError:Exception in the main thread java.lang.noClassDefFoundError: A at MainClass.main(MainClass.java:10) Caused by: java.lang.ClassNotFoundException: A at java.net.URLClassLoader.findClass(URLClassLoader.java.java <2> <3>:381) at java.lang.ClassLoader.loadClass(ClassLoader.java:424) at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331) on java.lang.ClassLoader.loadClass(ClassLoader.java:357) This can happen if:O file is not in the right directory. The class name must be the same as the file name (without the file extension). The names are sensitive to the case. Read this discussion about why NoClassDefFoundError occurs when running Java software. (@StackOverflow)25. NoSuchMethodFoundErrorThis error message will occur when java software tries to a class method and the method no longer has a definition definition The Java NoSuchMethodFoundError software error could not be found or loaded .java often occurs when there is a typo in the declaration. Read this tutorial to learn how to avoid the NoSuchMethodFoundError error message. (@javacodegeeks)26. NoSuchProviderExceptionNoSuchProviderException occurs when a security provider is prompted that is not available (@alvinalexander):javax.mail.noSuchProviderException When trying to figure out why NoSuchProviderException occurs, check:The JRE configuration. The Java house is defined in the configuration. Which Java environment is used. The security provider entry. Read this discussion about what Causes NoSuchProviderException when java software runs. (@StackOverflow)27. AccessControlExceptionAccessControlException indicates that requested access to system resources, such as a file system or network, is denied, as in this example of JBossDeveloper (@jbossdeveloper):ERROR Could not register java.security mbeans. AccessControlException: WFSM000001: Failed permission check (permission (javax.management.MBeanPermission org.apache.logging.log4j.core.jmx.LoggerContextAdmin#- [org.apache.logging.log4j2:type=51634f] registerMBean) in the code source (vfs:/C:/wildfly-10.0.0.Final/standalone/deployments/mySampleSecurityApp.war/WEB-INF/lib/log4j-core-2.5.jar) Read this discussion about a workaround used to overcome an AccessControlException error. (@github)28. ArrayStoreExceptionAn ArrayStoreException occurs when the rules of casting elements in Java arrays are broken. Arrays are very careful about what can get into them. (@Roedyg) For example, this example of JavaScan.com illustrates that this program (@java_scan): /* ..... START ..... */ public class JavaArrayStoreException { main public static void (String... args) { Object[] val = new Integer[4]; val[0] = 5.8; } /* ..... END ..... */ Results in the following output:Exception in the main thread java.lang.ArrayStoreException: java.lang.Double atHandling Exception.JavaArrayStoreException.main (JavaArrayStoreException.java:7) When an array is initialized, the types of objects allowed in the array need to be declared. Then, each array element must be of the same object type. Read this discussion on how to resolve for arraystoreexception. (@StackOverflow)29. Bad magic numberThis Java software error message means that something may be wrong with the class definition files on the network. Here is an example server side (@TSS_dotcom):Java(TM) Plug-in: Version 1.3.1_01 Using jre version 1.3.1_01 Java HotSpot (TM) Client VM user home directory = C:\Documents and Settings\Ankur Proxy Configuration: Manual configuration proxy: 192.168.11.6:80 SalesCalculatorAppletBeanInfo (Bad Magic Number) on in Method) in java.lang.ClassLoader.defineClass (Unknown Source) at java.security.SecureClassLoader.defineClass (Unknown Source) at sun.applet.AppletClassLoader.findClass (Unknown Source) at sun.plugin.security.PluginClassLoader.access$201 (Unknown Source) at sun.plugin.security.PluginClassLoader$1.run (Unknown Source) at java.security.AccessController.doController sun.plugin.security.PluginClassLoader.findClass (Unknown Source) at java.lang.ClassLoader.loadClass (Unknown Source) at sun.applet.AppletClassLoader.loadClass (Unknown Source) at java.lang.ClassLoader.loadClass (Unknown Source) at java.beans.Introspector.instantiate(Unknown Source) at java.beans.Introspector(Unknown Source) at java.beans.Introspector. (Unknown source) at java.beans.Introspector.getBeanInfo (Unknown Source) at sun.beans.ole.OleBeanInfo. (Unknown source) at sun.beans.ole.StubInformation.getStub (Unknown Source) at sun.plugin.ocx.TypeLibManager$1.run (Unknown Source) on java.security.AccessController.doPrivileged(Native Method) to the sun.plugin.ocx.TypeLibManager.getTypeLib (Unknown Source) to the sun.plugin.ocx.TypeLibManager.getTypeLib(Unknown Source) to the sun.plugin.ocx.ActiveXAppLet.statusNotification(Native Method) at sun.plugin.ocx.activeXAppletViewer.notifyStatus(Unknown Source) at sun.plugin.ocx.ActiveXAppletViewer.showAppletStatus(Unknown Source) at sun.applet.AppletPanel.run (Unknown Source) in java.lang.Thread.run(Unknown Source) The bad magic number error message can happen when:The first four bytes of a class file is not the CAFEBABE hexadecimal number. The class file was loaded as in ASCII mode, not binary mode. The java program runs before it is compiled. Read this thread on how to find the reason for a bad magic number. (@coderanch)30. Broken tubeThis error message refers to the data flow of a file or network outlet stopped working or is closed from the other end (@ExpertsExchange). Exception on the main line java.net.SocketException: Broken tube in java.net.SocketOutputStream.socketWrite0(Native Method) on java.net.SocketOutputStream.socketWrite (SocketOutputStream.java:92) in java.net.SocketOutputStream.write (SocketOutputStream.java:115) in java.io.DataStreamStream.write The causes of a broken pipe usually include:Running out of disk risk space. Ram may be clogged. The data flow can be corrupt. The pipe reading process may have been terminated. Read this thread about what is the Broken Tube Java error. (@StackOverflow)31. could not create the Java Virtual MachineThis Java error message usually occurs when code tries to invoke Java with the wrong arguments (@ghacksnews):Error: The Java Virtual Machine Error Could Not Be Created: One fatal occurred. The show's coming out. It is often caused by an error in the declaration in code or by assigned the appropriate amount of memory to it. Read this. Right. how to fix the Java software error I could not create the Java Virtual Machine.. (@StackOverflow)32. Class file contains wrong classThe wrong class file problem occurs when Java code tries to find the class file in the wrong directory, resulting in an error message similar to the following:MyTest.java:10: cannot access the bad class file MyStruct: D:\Java\test\MyStruct.java file does not contain MyStruct class Please remove or ensure that it appears in the correct classpath subdirectory. MyStruct ms = new MyStruct(); ^ To fix this error, these tips can help:Make sure that the name of the source file and the name of the class match—including the case. Make sure the package declaration is correct or missing. Make sure that the source file is in the right directory. Read this discussion on how to fix a class file error that contains wrong class. (@StackOverflow)33. ClassCastExceptionThe ClassCastException message indicates that Java code is trying to throw an object to the wrong class. In this example of the Java Concept of the Day, running the following program:package com; class A { int i = 10; } class B extends A { int j = 20; } class C extends B { int k = 30; } public class ClassCastExceptionDemo { main public static void (String[] args) { A a = new type B(); //B type is auto up thrown to a type B b = (B) a; //A type is explicitly casted for type B.C c = (C) b; Here, you'll have the exception of the System.out.println(c.k) class cast; } } Results in this error:Exception on the main line java.lang.ClassCastException: com. B cannot be released to com. C in with. ClassCastExceptionDemo.main (ClassCastExceptionDemo.java:23) Java code will create a hierarchy of classes and subclasses. To avoid the ClassCastException error, make sure that the new type belongs to the right class or one of its parent classes. If generics are used, these errors can be caught when the code is compiled. Read this tutorial on how to fix Java ClassCastException software errors. (@java_concept)34. ClassFormatErrorThe ClassFormatEror message indicates a binding error and occurs when a class file cannot be read or interpreted as a class file. Caused by: java.lang.ClassFormatEror: Missing code attribute in method that is not native or abstract in the javax/persistence/GenerationType class file in java.lang.ClassLoader.defineClass1(Native Method) in java.lang.ClassLoader.defineClassCond(Unknown Font) in java.lang.ClassLoader.defineClass(Unknown Source) in java.lang.Class net. URLClassLoader.defineClass at java.net.URLClassLoader.access$000 at java.net.URLClassLoader$1.run at java.security.AccessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) at java.security.accessController.doPrivileged(Native Method) Unknown) in java.lang.ClassLoader.loadClass(Unknown Source) in java. Java. Source) in java.lang.ClassLoader.loadClass (Unknown Source) There are several reasons why a ClassFormatError can occur:O class file was loaded as in ASCII mode not in binary mode. The web server should send class files as binary not ASCII. There may be a classpath error that prevents the code from finding the class file. If the class is loaded twice, the second time will cause the exception to be thrown. An old version of the Java runtime is being used. Read this discussion about what causes ClassFormatError in Java. (@StackOverflow)35. ClassNotFoundExceptionClassNotFoundException only occurs at runtime—which means that a class that was there during compilation is missing at run time. This is a linkage error. Like NoClassDefFoundError, this problem can occur if:O file is not in the right directory. The class name must be the same as the file name (without the file extension). The names are sensitive to the case. Read this discussion about what ClassNotFoundException causes for more cases. (@StackOverflow)36. ExceptionIninializerErrorThis Java problem will occur when something goes wrong with a static boot (@GitHub). When Java code later uses the class, noclassdeffounderror error will occur.java.lang.exceptionInitializerError at org.eclipse.mat.hprofIndexBuilder.fill(HprofIndexBuilder.java:54) at org.eclipse.mat.parser.internal.SnapshotFactory.parse(SnapshotFactory.java y <0>:193) at org.eclipse.mat.parser.internal.SnapshotFactory.openSnapshot(SnapshotFactory.java:106) at com.squareup.leakcanary.HeapAnalyzer.openSnapshot(HeapAnalyzer.java:134) at com.squareup.leakcanary.HeapAnalyzer.checkFor Leak(HeapAnalyzer.java:87) at com.squareup.leakcanary.internal.HeapAnalyzerService.onHandleCare(HeapAnalyzerService.java:56) on android.app.IntentService$ServiceHandler.handleMessage(IntentService.java:65) at android.os.Handler.dispatchMessage(Handler.java::102) on android Caused.java by: java.java.lang.NullPointerException: in == null at java.util.util.load(Properties.java:246) at org.eclipse.mat.util.MessageUtil. (MessageUtil.java:28) at org.eclipse.mat.util.MessageUtil. (MessageUtil.java:13) ... Plus 10 There needs to be more information to fix the error. Using getCause() in code can return the exception that caused the error to be returned. Read this discussion on how to track the cause of the Excededor. (@StackOverflow)37. IllegalBlockSizeExceptionAn IllegalBlockSizeException will occur during decryption when the length message is not an 8-byte multiple. Here is an example of ProgramCreek.com (@ProgramCreek):@Override byte protected[] engineWrap(keykey) launches IllegalBlockSizeException, InvalidKeyException { try encoded = key.getEncoded(); return return 0, encoded.length); } catch (BadPaddingException e) { IllegalBlockSizeException newE = new IllegalBlockSizeException(); newE.initCause(e); throw newE; } } The IllegalBlockSizeException can be caused by:Different encryption options and decryption algorithm used. The message to be decrypted can be truncated or shuffled in the transmission. Read this discussion on how to avoid the IllegalBlockSizeException Java software error message. (@StackOverflow)38. BadPaddingExceptionA BadPaddingException will occur during decryption when padding was used to create a message that can be measured by a multiple of 8 bytes. Here's an example of Stack Overflow (@StackOverflow):javax.crypto.BadPaddingException: Given the final block not properly padded in com.sun.crypto.provider.SunJCE_f.b(DashoA13*.) at com.sun.crypto.provider.SunJCE_f.b(DashoA113*.) at com.sun.crypto.provider.AESCipher.engineDoFinal(DashoA13*.) at javax.crypto.Cipher.doFinal(DashoA13*.) Encrypted data is binary, so do not attempt to store it in a sequence or the data has not been padded correctly during encryption. Read this discussion on how to prevent BadPaddingException. (@StackOverflow)39. IncompatibleClassChangeErrorAn IncompatibleClassChangeError is a form of LinkageError that can occur when a base class changes after compiling a child class. This example is from How to Do in Java (@HowToDoInJava):Exception in thread main java.lang.IncompatibleClassChangeError: Implementing class at java.lang.ClassLoader.defineClass1(Native Method) at java.lang.ClassLoader.defineClass(Unknown Source) at java.security.URLClassLoader.$00(unknown Source) at java.net.URLClassLoader.access Unknown Source) at java.net.URLClassLoader$1.run(Unknown Source) at java.security.AccessController.doPrivileged(Native Method) at java.net.URLClassLoader.findClass(Unknown Source) at java.lang.loadClass(Unknown Source) at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source) at java.lang.ClassLoader.loadClassInternal(Unknown Source) at net.sf.cglib.core.DebuggingClassWriter.toByteArray(DebuggingClassWriter.java:73) at net.sf.cglib.core.DefaultGeneratorStrategy.generate(DefaultGeneratorStrategy.java:26) at net.sf.cglib.core.AbstractClassGenerator.create(AbstractClassGenerator.java:216) at net.sf.cglib.core.core.core.core.create(AbstractClassGeneratorgenerator.java:216) at net.sf.cglib.core.core.core.core.create(AbstractClassGeneratorgenerator.java:216) at net.sf.cglib.core.core.core.core KeyFactory$Generator.create(KeyFactory.java:144) at net.sf.cglib.core.KeyFactory.create(KeyFactory.java:116) at net.sf.cglib.coreFactory.create(KeyFactory.java:108) at net.sf.cglib.core.KeyFactory.create(KeyFactory.java:104) at net.sf.cglib.proxy.Enhancer. (Enhancer.java:69) When the IncompatibleClassChangeError occurs, it is possible that the static in the main method has been A legal class was used illegally. A lesson was changed and are references to it from another class by its old signatures. Try deleting all class files and recompiling everything. Try these steps to resolve the IncompatibleClassChangeError. (@javacodegeeks)40. FileNotFoundExceptionThis Java software error message is released when a file with the specified path name does not exist.@Override the public file ParcelFileDescriptor openFile(Uri uri,String mode) launches FileNotFoundException { if (uri.toString()(FILE_PROVIDER_PREFIX)) { int m=ParcelFileDescriptor.MODE_READ_ONLY; if (mode.equalsIgnoreCase(rw)) m=ParcelFileDescriptor.MODE_READ_WRITE; File f=new File(uri.getPath()); ParcelFileDescriptor pfd=ParcelFileDescriptor.open(f,m); pfd return; } another { relaunch new FileNotFoundException(Unupported uri: + uri.toString()); } } In addition to files that do not exist the specified path name, this may mean that the existing file is inaccessible. Read this discussion on why FileNotFoundException can be released. (@StackOverflow)41. EOFExceptionAn EOFException is released when a file end or end of the stream was reached unexpectedly during input. Here is an example of JavaBeat from an application that launches an EOFException:import java.io.DataInputStream; import java.io.EOFException; import java.io.File; import java.io.FileInputStream; import java.io.IOException; exception of the public classExample { public void testMethod1(){ File file = new file (test.txt); DataInputStream dataInputStream = null; try{ dataInputStream = new DataInputStream (new fileFileInputStream(file)); while (true){ dataInputStream.readInt(); } catch (EOFException e){ e.printStackTrace(); } catch (IOException e){ e.printStackTrace(); } finally try{ { if (dataInputStream != null){ dataInputStream.close(); } capture (IOException e){ e.printStackTrace(); } } main public static void (String[] args){ ExceptionExample instance1 = new ExceptionExample(); instance1.testMethod1(); } } Running the above program results in the following exception:java.io.EOFException in java.io.DataInputStream.readInt(DataInputStream.java:392) in logging.simple.ExceptionExample.testMethod1(ExceptionExample.java .java:16) in logging.simple.ExceptionExample.main (ExceptionExample.java:36) When there is no more data while the Class DataInputStream is trying to read data in the stream, EOFException exception is released. It can also occur in the ObjectInputStream and RandomAccessFile classes. Read this discussion about when EOFException may occur while running Java software. (@StackOverflow)42. UnupportedEncodingExceptionThis Java software error message is released when character encoding is not supported (@Penn).public UnupportedEncodingException() It is possible that the Java Virtual Machine being used does not support a certain of characters. Read this discussion on how to handle UnupportedEncodingException while running Java software. Java. SocketExceptionA SocketException indicates that there is an error creating or accessing a socket (@ProgramCreek). Sequence period - getAttribute(PERIOD_PROPERTY); if (periodStr != null) { period int = 0; try { period = Integer.parseInt(periodStr); } catch (NumberFormatException nfe) { } if (period <= 0) { launch new metricsException(Invalid period: + periodStr); } setPeriod(period); } metricsServers = Util.parse (get(SERVERS_PROPERTY), DEFAULT_PORT); unitsTable = getAttributeTable (UNITS_PROPERTY); slope = getAttributeTable (SLOPE_PROPERTY); tmaxTable = getAttributeTable (TMAX_PROPERTY); dmaxTable = getAttributeTable (DMAX_PROPERTY); try { datagramSocket = new DatagramSocket(); } catch (SocketException if) { se.printStackTrace(); } This exception is usually thrown when maximum connections are reached due to: No more network ports are available to the application. The system does not have enough memory to support new connections. Read this discussion on how to resolve SocketException issues while running Java software. (@StackOverflow)44. SSLExceptionThis Java software error message occurs when SSL-related operations fail. The following example is from Atlassian (@Atlassian):com.sun.jersey.api.client.ClientHandlerException: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error: java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty at com.sun.jersey.client.apache.ApacheHttpClientHandler.handle(ApacheHttpClientHandler.java:202) at com.sun.jersey.api.client.Client.handle(Client.java:365) at com.sun.jersey.api.client.WebResource.handle(WebResource.java:556) at com.sun.jersey.api.client.WebResource.get(WebResource.java:178) at com.atlassian.plugins.client.service.product.ProductServiceClientImpl.getProductVersionsAfterVersion(ProductServiceClientImpl.java:82) at com.atlassian.upm.pac.PacClientImpl.getProductUpgrades(PacClientImpl.java:111) at com.atlassian.upm.rest.resources.ProductUpgradesResource.get(ProductUpgradesResource.java:39) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source) at java.lang.reflect.Method.invoke(Unknown Source) at com.atlassian.plugins.rest.common.interceptor.impl.DispatchProviderHelper$ResponseOutInvoker$1.invoke(DispatchProviderHelper.java:206) at com.atlassian.plugins.rest.common.interceptor.impl.DispatchProviderHelper$1.intercept(DispatchProviderHelper.java:90) at com.atlassian.plugins.rest.common.interceptor.impl.DefaultMethodInvocation.invoke(DefaultMethodInvocation.java:61) at (DefaultMethodInvocation.java:61) at com.atlassian.plugins.rest.resordente.common.interceptor.impl.DispatchProviderHelper.invokeMethodWithInterceptors(DispatchViderHelper.java:98) at com.atlassian.plugins.plugins.rest.common.interceptor.impl.DispatchProviderHelper.access$100(DispatchProviderHelper.java:28) at com.atlassian.plugins.rest.common.interceptor.impl.DispatchHelperProvider$ResponseOutInvoker._dispatch (DispatchProviderHelper.java:202) ... Caused by: javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error: java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be not empty ... Caused by: java.lang.RuntimeException: Unexpected error: java.security.InvalidParameterParameterException: the trustAnchors parameter must be non-empty ... Caused by: java.security.InvalidAlgorithmParameterException: The trustAnchors parameter must be non-empty This can happen if:Certificates on the server or client have expired. The server port has been reset to another port. Read this discussion about what may cause the SSLException error in java software. (@StackOverflow)45. MissingResourceExceptionA MissingResourceException occurs when a resource is missing. If the resource is in the correct classpath, this is usually because a property file is not configured correctly. Here is an example (@TIBCO):java.util.MissingResourceException: Cannot find package for base name localemsgs_en_US, locale en_US java.util.ResourceBundle.throwMissingResourceException java.util.ResourceBundle.getBundleImpl java.util.ResourceBundle.. ResourceBundle.getBundle net.sf.jasperreports.engine.util.JRResourcesUtil.loadResourceBundle net.sf.jasperreports.engine.util.util.loadResourceResourceBundle Read this discussion on how to fix MissingResourceException while running Java.46 software. NoInitialContextExceptionThe NoInitialContextException occurs when the Java application wants to perform a naming operation, but cannot create a connection (@TheASF). [java] Caused by: javax.naming.NoInitialContextException: Need to specify the name of the class in system environment or property, or as an applet parameter, or in an application resource file: java.naming.factory.initial [java] at javax.naming.spi.NamingManager.getInitialContext(NamingManager.java:645) [java] in javax.naming.InitialContext.getDefaultInitCtx (InitialContext.java:247) [java] in javax.naming.InitialURLOrDefaultInitCtx (InitialContext.java:284) [java] at javax.naming.Context.lookup(InitialContext.java:351) [java] at org.apache.camel.impl.JndiRegistry.lookup(JndiRegistry.java:51) This may be a complex problem to solve, but here are some possible problems that cause the Java NoInitialContextException:O application may not have the appropriate credentials to make a connection. code can't can implementation of the jndi required. The InitialContext class cannot be configured with the right properties. Read this discussion about what NoInitialContextException means when running Java software. (@StackOverflow)47. NoSuchElementExceptionA NoSuchElementException happens when an iteration (such as a for loop tries to access the next element when there is no.public class NoSuchElementExceptionDemo{ public static void main (String args[]) { Hashtable sampleMap = new Hashtable(); Enumeration enumeration enumeration = sampleMap.elements(); enumeration.nextElement(); java.util.NoSuchElementExcepiton here because the enumeration is empty } } Output: Exception on the main thread java.util.NoSuchElementException: Hashtable Enumerator at java.util.Hashtable$EmptyEnumerator.nextElement(Hashtable.java:1084) under test. ExceptionTest.main(NoSuchElementExceptionDemo.java:23) NoSuchElementException can be released by these methods:Enumeration::nextElement()NamingEnumeration::next()StringTokenizer::nextElement()Iterator:next()Read this tutorial on how to fix NoSuchElementexception in Java software. (@javinpaul)48. NoSuchFieldErrorThis Java software error message is released when an application tries to access a field on an object, but the specified field no longer exists in the onbject (@sourceforge).public NoSuchFieldError() Usually this error is caught in the compiler, but will be caught during runtime if a class definition has been changed between compiling and executing. Read this discussion on how to find what causes NoSuchFieldError when running Java software. @StackOverflow49. NumberFormatExceptionThis Java software error message occurs when the application tries to convert a string to a numeric type, but that number is not a valid string of digits (@alvinalexander.package com.devdaily.javasamples; public class ConvertStringToNumber { public static void main (String[] args) { try { String s = FOOBAR; int i = Integer.parseInt(s); // this line of code will never be reached System.out.println(int value = + i); } capture (NumberFormatException nfe) { nfe.printStackTrace(); } } The NumberFormatException can be released when:leading or following spaces in the number. The signal's not in front of the number. The number has compulsion. The location may not classify it as a valid number. The number is too large to fit the numeric type. Read this discussion on how to avoid NumberFormatException when running Java software. (@StackOverflow)50. TimeoutExceptionThis Java software error message occurs when a lock operation runs out.Private empty queueObject (ComplexDataObject obj) plays TimeoutException, InterruptedException { if (!queue.offer(obj,10,TimeUnit.SECONDS){ TimeoutException ex=new TimeoutException(Timed waiting for analyzed elements to be processed. launch ex; } } Read this discussion about to handle TimeoutException when running Java software. (@StackOverflow) For the best Java developer toolkit, be sure to download the Java Developer Comprehensive Guide. Guide.