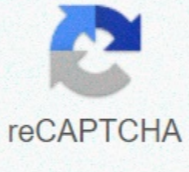




I'm not robot



Continue

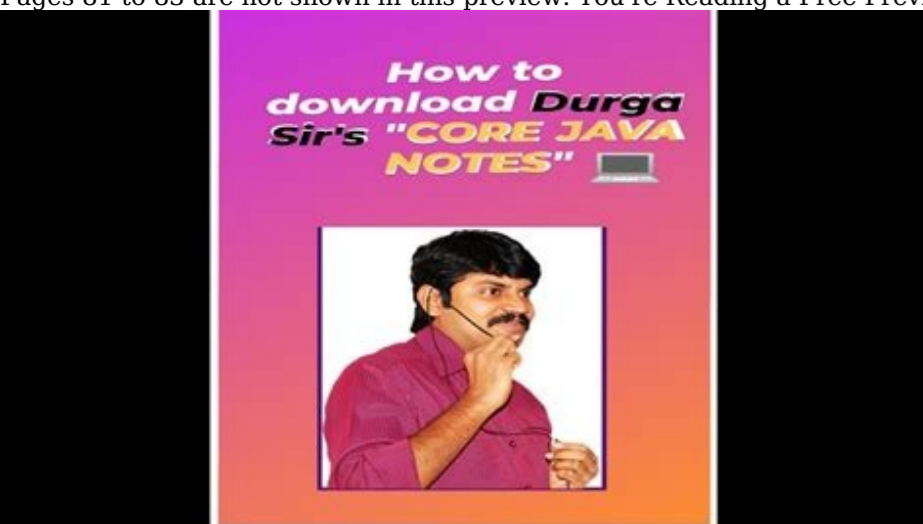
Durga sir core java pdf

Core java with scjp/ ocjp notes by durga sir pdf. Durga sir core java notes pdf free download.

Type: PDF Date: May 2020 Size: 18.5MB Author: intjar malik This document was uploaded by user and they confirmed that they have the permission to share it. If you are author or own the copyright of this book, please report to us by using this DMCA report form.



Report DMCA DOWNLOAD PDF This is a non-profit website to share the knowledge. To maintain this website, we need your help. A small donation will help us alot. You're Reading a Free Preview Pages 16 to 19 are not shown in this preview. You're Reading a Free Preview Pages 31 to 66 are not shown in this preview. You're Reading a Free Preview Pages 81 to 83 are not shown in this preview. You're Reading a Free Preview Pages 87 to 89 are not shown in this preview. You're Reading a Free Preview Pages 93 to 102 are not shown in this preview. You're Reading a Free Preview Pages 106 to 112 are not shown in this preview.



You're Reading a Free Preview Pages 116 to 120 are not shown in this preview. You're Reading a Free Preview Pages 156 to 177 are not shown in this preview. You're Reading a Free Preview Pages 202 to 272 are not shown in this preview. You're Reading a Free Preview Pages 319 to 371 are not shown in this preview. You're Reading a Free Preview Pages 396 to 485 are not shown in this preview. You're Reading a Free Preview Pages 507 to 554 are not shown in this preview. You're Reading a Free Preview Pages 559 to 561 are not shown in this preview. You're Reading a Free Preview Pages 566 to 572 are not shown in this preview. No Text Content! total; } public static void main(String[] args) { sum(); sum(10); sum(10,20); sum(10,20,30,40); } } Output: The sum: 0 The sum: 10 The sum: 30 The sum: 100 Case 1: Which of the following var-arg method declarations are valid? 1. methodOne(int... x) (valid) 2. methodOne(int ...x) (valid) 3. methodOne(int...x) (valid) 4. methodOne(int x...) (invalid) 5. methodOne(int...x) (invalid) 6. methodOne(int x...) (invalid) Case 2: We can mix var-arg parameter with general parameters also. Example: methodOne(int a,int... b)//valid methodOne(String s,int... x)//valid Case 3: if we mix var-arg parameter with general parameter then var-arg parameter should be 42 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals public void methodOne(int... i) } } Output: Compile time error: Cannot declare both methodOne(int...) and methodOne(int[]) in Test Single Dimensional Array Vs Var-Arg Method: Case 1: Wherever single dimensional array present we can replace with var-arg parameter. Example: class Test { public static void main(String... args) { System.out.println("var-arg main method");//var-arg main method } } Case 2: Wherever var-arg parameter present we can't replace with single dimensional array. Note : 1. methodOne(int... x) we can call this method by passing a group of int values and x will become 1D array. (i.e., int[] x) 2. methodOne(int[]... x) we can call this method by passing a group of 1D int[] and x will become 2D array. (i.e., int[][] x) Above reasons this case 2 is invalid. Example: class Test { public static void methodOne(int[]... x) { for(int[] a:x) { System.out.println(a[0]); 44 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals } } public static void main(String[] args) { int[] l={10,20,30}; int[] m={40,50}; methodOne(l,m); } } Output: 10 40 Analysis: Main Method Whether the class contains main() method or not, and whether it is properly declared or not, these checkings are not responsibilities of the compiler, at runtime JVM is responsible for this. If JVM unable to find the required main() method then we will get runtime exception saying NoSuchMethodError: main. Example: class Test { } Output: javac Test.java java Test R.E: NoSuchMethodError: main At runtime JVM always searches for the main() method with the following prototype. 45 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals If we are performing any changes to the above syntax then the code won't run and will get Runtime exception saying NoSuchMethodError. Even though above syntax is very strict but the following changes are acceptable to main() method. 1. The order of modifiers is not important that is instead of public static we can take static public. 2. We can declare string[] in any acceptable form o String[] args o String []args o String args[] 3. Instead of args we can use any valid java identifier. 4. We can replace string[] with var-arg parameter. Example: main(String... args) 5. main() method can be declared with the following modifiers. final, synchronized, strictfp. 6. class Test { 7. static final synchronized strictfp public void main(String... ask){ 8. System.out.println("valid main method"); 9.



a,int b) //(invalid) Case 4: With in the var-arg method we can take only one var-arg parameter. i.e., if we are trying to more than one var-arg parameter we will get CE. Example: methodOne(int... a,int... b) //(invalid) Case 5: class Test { public static void methodOne(int i) { System.out.println("general method"); } public static void methodOne(int... i) { System.out.println("var-arg method"); } } public static void main(String[] args) { methodOne();//var-arg method methodOne(10,20);//general method } } In general var-arg method will get least priority that is if no other method matched then only var-arg method will get the chance this is exactly same as default case inside a switch. Case 6: For the var-arg methods we can provide the corresponding type array as argument. Example: class Test { { System.out.println("var-arg method"); } public static void main(String[] args) { methodOne(new int[] {10,20,30});//var-arg method } } Case 7: class Test { public void methodOne(int[] i){ 43 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals public void methodOne(int... i) } } Output: Compile time error: Cannot declare both methodOne(int...) and methodOne(int[]) in Test Single Dimensional Array Vs Var-Arg Method: Case 1: Wherever single dimensional array present we can replace with var-arg parameter. Example: class Test { public static void main(String... args) { System.out.println("var-arg main method");//var-arg main method } } Case 2: Wherever var-arg parameter present we can't replace with single dimensional array. Note : 1. methodOne(int... x) we can call this method by passing a group of int values and x will become 1D array. (i.e., int[] x) 2. methodOne(int[]... x) we can call this method by passing a group of 1D int[] and x will become 2D array. (i.e., int[][] x) Above reasons this case 2 is invalid. Example: class Test { public static void methodOne(int[]... x) { for(int[] a:x) { System.out.println(a[0]); 44 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals } } public static void main(String[] args) { int[] l={10,20,30}; int[] m={40,50}; methodOne(l,m); } } Output: 10 40 Analysis: Main Method Whether the class contains main() method or not, and whether it is properly declared or not, these checkings are not responsibilities of the compiler, at runtime JVM is responsible for this. If JVM unable to find the required main() method then we will get runtime exception saying NoSuchMethodError: main. Example: class Test { } Output: javac Test.java java Test R.E: NoSuchMethodError: main At runtime JVM always searches for the main() method with the following prototype. 45 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals If we are performing any changes to the above syntax then the code won't run and will get Runtime exception saying NoSuchMethodError. Even though above syntax is very strict but the following changes are acceptable to main() method. 1. The order of modifiers is not important that is instead of public static we can take static public. 2. We can declare string[] in any acceptable form o String[] args o String []args o String args[] 3. Instead of args we can use any valid java identifier. 4. We can replace string[] with var-arg parameter. Example: main(String... args) 5. main() method can be declared with the following modifiers. final, synchronized, strictfp. 6. class Test { 7. static final synchronized strictfp public void main(String... ask){ 8. System.out.println("valid main method"); 9.



} 10. } 11. output : 12.

CONTENTS:

- 1. Language Fundamentals
- 2. Operators and Assignments
- 3. Declaration and Access Control
- 4. Flow Control
- 5. Exception Handling
- 6. Assertions
- 7. OO Concepts
- 8. Inner Classes
- 9. Threads and Concurrency
- 10. Fundamental classes in java.lang package
 - 1. Object Class
 - 2. String class
 - 3. StringBuffer Class
 - 4. StringBuilder Class
 - 5. Wrapper Classes
 - 6. Math class
- 11. The Collections frame work and Generics
- 12. File IO & Serialization
- 13. Garbage Collection
- 14. 1.5 version New Features
 - 1. Enum
 - 2. For-Each Loop
 - 3. Var-Arg Methods
 - 4. Auto Boxing & Un Boxing
 - 5. Static Imports
- 15. Internationalization
- 16. QuickReference

1

LANGUAGE FUNDAMENTALS



valid main method Which of the following main() method declarations are valid ? 1. public static void main(String args){ } (invalid) 2. public synchronized final strictfp void main(String[] args){ } (invalid) 3. public static void Main(String... args){ } (invalid) 4. public static int main(String[] args){ } //int return type we can't take //(invalid) 5. public static synchronized final strictfp void main(String... args){ } (valid) 6. public static void main(String... args){ } (valid) 7. public void main(String[] args){ } (invalid) In which of the above cases we will get compile time error ? No case, in all the cases we will get runtime exception. Case 1 : Overloading of the main() method is possible but JVM always calls string[] argument main() method only.

46 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals Example: class Test //overloaded { public static void main(String[] args) { System.out.println("String[] array main method"); } } public static void main(int[] args) { System.out.println("int[] array main method"); } } Output: String[] array main method The other overloaded method we have to call explicitly then only it will be executed. Case 2: Inheritance concept is applicable for static methods including main() method hence while executing child class if the child class doesn't contain main() method then the parent class main() method will be executed. Example 1: //Parent.java class Parent { public static void main(String[] args) { System.out.println("parent main"); } } class Child extends Parent { } 47 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals Analysis: Example 2: // Parent.java class Parent { public static void main(String[] args) { System.out.println("parent main"); } } class Child extends Parent { public static void main(String[] args) { System.out.println("Child main"); } } 48 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals Analysis: It seems to be overriding concept is applicable for static methods but it is not overriding it is method hiding. 1.7 Version Enhancements with respect to main(): Case 1 : Untill 1.6 if our class doesn't contain main() method then at runtime we will get Runtime Exception saying NoSuchMethodError:main But from 1.7 version onwards instead of NoSuchMethodError we will get more meaning full description class Test { } 1.6 version : javac Test.java java Test RE: NoSuchMethodError:main 1.7 version : define the main method javac Test.java java Test Error: main method not found in class Test, please as public static void main(String[] args) 49 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Core Java with SCJP/ OCJP Notes By Durga Sir Language Fundamentals Case 2 : From 1.7 version onwards to start program execution compulsory main method should be required, hence even though the class contains static block if main method not available then won't be executed class Test { static { System.out.println("static block"); } } 1.6 version : javac Test.java java Test output : static block RE: NoSuchMethodError:main 1.7 version : define the main method javac Test.java java Test Error: main method not found in class Test, please as public static void main(String[] args) Case 3 : class Test { static { System.out.println("static block"); } System.exit(0); } } 1.6 version : javac Test.java java Test output : static block 1.7 version : define the main method javac Test.java java Test Error: main method not found in class Test, please as public static void main(String[] args) Case 4 : class Test { static { System.out.println("static block"); } public static void main(String[] args) { System.out.println("main method"); } } 1.6 version : javac Test.java java Test 50 DURGASOFT, # 202,2ndFloor,HUDA Maitrivanam,Ameerpet, Hyderabad - 500038, 040 - 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com Academia.edu uses cookies to personalize content, tailor ads and improve the user experience. By using our site, you agree to our collection of information through the use of cookies. To learn more, view our Privacy Policy.