Top of Form

|  |  |
| --- | --- |
| **1.** | import java.io.File;  public class M1 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello1.txt");  f1.createNewFile();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  RuntimeException |  |  | | --- | | C.  None | | | |
|  | | |
| **2.** | import java.io.File;  public class M2 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello2.html");  f1.createNewFile();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  RuntimeException |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **3.** | import java.io.File;  public class M3 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello3.doc");  f1.createNewFile();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  RuntimeException |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |
| **4.** | import java.io.File;  public class M4 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello4.java");  boolean flag = f1.createNewFile();  System.out.println("done:" + flag);  }  }  // assume you are running first time the what could be the output |
| |  | | --- | | A.  done: true |  |  | | --- | | B.  done: false |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **5.** | import java.io.File;  public class M4 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello4.java");  boolean flag = f1.createNewFile();  System.out.println("done:" + flag);  }  }  // assume you are running second time the what could be the output |
| |  | | --- | | A.  done: true |  |  | | --- | | B.  done: false |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |
| **6.** | import java.io.File;  public class M5 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello5.txt");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.createNewFile());  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a:true  b:false  c:true  done: |  |  | | --- | | B.  a: false  b:true  c: true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **7.** | import java.io.File;  public class M5 {  public static void main(String[] args) throws Exception {  File f1 = new File("hello5.txt");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.createNewFile());  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a:true  b:false  c:true  done: |  |  | | --- | | B.  a: false  b:true  c: true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |
| **8.** | import java.io.File;  import java.io.IOException;  public class M6 {  public static void main(String[] args) {  File f1 = new File("hello6.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:true  c:true  done: |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **9.** | import java.io.File;  import java.io.IOException;  public class M6 {  public static void main(String[] args) {  File f1 = new File("hello6.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output | |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:true  c:true  done: |  |  | | --- | | D.  None | | | | |
|  | | | |
| **10.** | | import java.io.File;  import java.io.IOException;  public class M7 {  public static void main(String[] args) {  File f1 = new File("E:\\hello7.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: true  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | | |
|  | | | |

|  |  |
| --- | --- |
| **11.** | import java.io.File;  import java.io.IOException;  public class M7 {  public static void main(String[] args) {  File f1 = new File("E:\\hello7.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  None | | | |
|  | | |
| **12.** | import java.io.File;  import java.io.IOException;  public class M8 {  public static void main(String[] args) {  File f1 = new File("E:/hello8.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  a:false  b:true  c:true  done: | | | |
|  | | |

|  |  |
| --- | --- |
| **13.** | import java.io.File;  import java.io.IOException;  public class M8 {  public static void main(String[] args) {  File f1 = new File("E:/hello8.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  compile Time Error |  |  | | --- | | D.  a:false  b:true  c:true  done: | | | |
|  | | |
| **14.** | import java.io.File;  import java.io.IOException;  public class M9 {  public static void main(String[] args) {  File f1 = new File("E://ocjp", "hello9.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:true  c:true  done: | | | |
|  | | |

|  |  |
| --- | --- |
| **15.** | import java.io.File;  import java.io.IOException;  public class M9 {  public static void main(String[] args) {  File f1 = new File("E://ocjp", "hello9.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  false  c:false  done: | | | |
|  | | |
| **16.** | import java.io.File;  import java.io.IOException;  public class M10 {  public static void main(String[] args) {  File dir = new File("E://ocjp");  File f1 = new File(dir, "hello10.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:false  b:true  c:true  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: | | | |
|  | | |

|  |  |
| --- | --- |
| **17.** | import java.io.File;  import java.io.IOException;  public class M10 {  public static void main(String[] args) {  File dir = new File("E://ocjp");  File f1 = new File(dir, "hello10.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  none | | | |
|  | | |
| **18.** | import java.io.File;  import java.io.IOException;  public class M11 {  public static void main(String[] args) {  File dir = new File("E://ocjp1");  File f1 = new File(dir, "hello11.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:false  b:true  c:true  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: | | | |
|  | | |

|  |  |
| --- | --- |
| **19.** | import java.io.File;  import java.io.IOException;  public class M11 {  public static void main(String[] args) {  File dir = new File("E://ocjp1");  File f1 = new File(dir, "hello11.txt");  System.out.println("a:" + f1.exists());  try  {  System.out.println("b:" + f1.createNewFile());  }  catch(IOException ex)  {  ex.printStackTrace();  }  System.out.println("c:" + f1.exists());  System.out.println("done:");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  none | | | |
|  | | |
| **20.** | import java.io.File;  public class M12 {  public static void main(String[] args) {  File f1 = new File("dir1");  f1.mkdir();  System.out.println("done");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  done |  |  | | --- | | B.  RuntimeException |  |  | | --- | | C.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **21.** | import java.io.File;  public class M12 {  public static void main(String[] args) {  File f1 = new File("dir1");  f1.mkdir();  System.out.println("done");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  done |  |  | | --- | | B.  RuntimeException |  |  | | --- | | C.  None | | | |
|  | | |
| **22.** | import java.io.File;  public class M13 {  public static void main(String[] args) {  File f1 = new File("dir2");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume that you are running first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: true  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  none | | | |
|  | | |

|  |  |
| --- | --- |
| **23.** | import java.io.File;  public class M13 {  public static void main(String[] args) {  File f1 = new File("dir2");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume that you are running second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  none | | | | |
|  | | | |
| **24.** | import java.io.File;  public class M14 {  public static void main(String[] args) {  File f1 = new File("D:/java8.0/dir3");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the first time what could be the output | |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | | |
|  | | | |

|  |  |
| --- | --- |
| **25.** | import java.io.File;  public class M14 {  public static void main(String[] args) {  File f1 = new File("D:/java8.0/dir3");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | |
|  | | |
| **26.** | import java.io.File;  public class M15 {  public static void main(String[] args) {  File f1 = new File("D:/java8.0/bin", "dir4");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the first time time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **27.** | import java.io.File;  public class M15 {  public static void main(String[] args) {  File f1 = new File("D:/java8.0/bin", "dir4");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the second time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | |
|  | | |
| **28.** | import java.io.File;  public class M16 {  public static void main(String[] args) {  File dir = new File("D:/java8.0/bin");  File f1 = new File(dir, "dir6");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the first time time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **29.** | import java.io.File;  public class M16 {  public static void main(String[] args) {  File dir = new File("D:/java8.0/bin");  File f1 = new File(dir, "dir6");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the second time time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:false  c:false  done |  |  | | --- | | D.  a:false  b:true  c:true  done: |  |  | | --- | | E.  None | | | |
|  | | |
| **30.** | import java.io.File;  public class M17 {  public static void main(String[] args) {  File dir = new File("D:/java8.0/xyz");  File f1 = new File(dir, "dir6");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the first time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:true  c:true  done: |  |  | | --- | | D.  none | | | |
|  | | |

|  |  |
| --- | --- |
| **31.** | import java.io.File;  public class M17 {  public static void main(String[] args) {  File dir = new File("D:/java8.0/xyz");  File f1 = new File(dir, "dir6");  System.out.println("a:" + f1.exists());  System.out.println("b:" + f1.mkdir());  System.out.println("c:" + f1.exists());  System.out.println("done");  }  }  // assume this path is available in your system then you run the program for the second time time what could be the output |
| |  | | --- | | A.  a: false  b:true  c: false  done: |  |  | | --- | | B.  a:true  b:false  c:true  done: |  |  | | --- | | C.  a:false  b:true  c:true  done: |  |  | | --- | | D.  none | | | |
|  | | |
| **32.** | import java.io.File;  public class M18 {  public static void main(String[] args) {  File f1 = new File("hello1.txt");  File f2 = new File("dir1");  System.out.println("f1.isFile():" + f1.isFile());  System.out.println("f2.isFile():" + f2.isFile());  System.out.println("f1.isDirectory:" + f1.isDirectory());  System.out.println("f2.isDirectory:" + f2.isDirectory());  }  } |
| |  | | --- | | A.  f1.isFile():true  f2.isFile():false  f1.isDirectory:false  f2.isDirectory:true |  |  | | --- | | B.  f1.isFile():false  f2.isFile():true  f1.isDirectory:true  f2.isDirectory:false |  |  | | --- | | C.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **33.** | import java.io.File;  public class M19 {  public static void main(String[] args) {  File f1 = new File("D:\\java8.0\\bin");  File[] members = f1.listFiles();  for(File file : members)  {  System.out.println(file.getName());  }  }  }  // assume this path is available in your system then you run the program then you will get list of files names or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **34.** | import java.io.File;  public class M20 {  public static void main(String[] args) {  File src = new File("F:\\may7th");  File[] members = src.listFiles();  for(File member : members)  {  System.out.println(member.getName() + ":" + (member.isFile() ? " file " : " directory")) ;  }  }  }  // assume this path is available in your system then you run the program then you will get list of files names or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **35.** | import java.io.File;  public class M21 {  public static void main(String[] args) {  File src = new File("F:\\may7th");  File[] members = src.listFiles();  int dirCount = 0, fileCount = 0;  for(File member : members)  {  if(member.isFile())  {  fileCount ++;  }  else  {  dirCount ++;  }  }  System.out.println("files count:" + fileCount) ;  System.out.println("dir count:" + dirCount);  }  }  // assume this path is available in your system then you run the program then you will get count of files and directories or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
|  | | | |
| **36.** | import java.io.File;  import java.util.HashMap;  import java.util.Scanner;  public class M22 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("Enter source path");  String src = sc.next();  File srcDir = new File(src);  HashMap<String, Integer> map = new HashMap<String, Integer>();  File[] srcDirMembers = srcDir.listFiles();    String fileType;  for(File file : srcDirMembers)  {  if(file.isFile())  {  fileType = file.getName().substring(file.getName().indexOf(".") + 1);  if(map.get(fileType) == null)  {  map.put(fileType, 1);  }  else  {  map.put(fileType, map.get(fileType) + 1);  }  }  System.out.println(file.getName() + ":" + map);  }  System.out.println("final map:" + map);  System.out.println("total members:" + srcDirMembers.length);  }  }  // assume  F:\may7th  you give the input of this path and is available in your system then you run the program then you will get all the files and directories or not? | |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **37.** | import java.io.File;  import java.util.HashMap;  import java.util.Scanner;  public class M23 {    static HashMap<String, Integer> map = new HashMap<String, Integer>();  static HashMap<String, Integer> dirMap = new HashMap<String, Integer>();    public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("Enter source path");  String src = sc.next();  File srcDir = new File(src);  File[] srcDirMembers = srcDir.listFiles();  String fileType;  for(File file : srcDirMembers)  {  if(file.isFile())  {  countFile(file);  }  else  {  dirMap = new HashMap<String, Integer>();  processDir(file);  System.out.println(file.getName() + ":" + dirMap);  }  }  System.out.println(srcDir + ":" + map);  }  static void countFile(File f1)  {  String fileType = f1.getName().substring(f1.getName().indexOf(".") + 1);  if(map.get(fileType) == null)  {  map.put(fileType, 1);  }  else  {  map.put(fileType, map.get(fileType) + 1);  }  if(dirMap.get(fileType) == null)  {  dirMap.put(fileType, 1);  }  else  {  dirMap.put(fileType, dirMap.get(fileType) + 1);  }  }  static void processDir(File dir)  {  File[] srcDirMembers = dir.listFiles();  for(File file : srcDirMembers)  {  if(file.isFile())  {  countFile(file);  }  else  {  processDir(file);  }  }  }  }  // assume  F:\may7th  you give the input of this path and is available in your system then you run the program then you will get all the directories with directories and files inside it or not? | |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
|  | | | |
| **38.** | package io2;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  public class M1 {  public static void main(String[] args) throws IOException{  File f1 = new File("test1.txt");  FileWriter out = new FileWriter(f1);  out.write("Hello");  out.write("test");  out.write("abc");  out.write("xyz");  out.write("end");  out.flush();  out.close();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  throws Exception | | | | |
|  | | | |

|  |  |
| --- | --- |
| **39.** | package io2;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  public class M2 {  public static void main(String[] args) throws IOException{  File f1 = new File("test2.txt");  FileWriter out = new FileWriter(f1);  out.write("Hello");  out.write("test");  out.write("abc");  out.write("\n");  out.write("xyz");  out.write("end");  out.flush();  out.close();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  throws Exception | | | | |
|  | | | |
| **40.** | package io2;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  public class M3 {  public static void main(String[] args) throws IOException{  FileWriter out = new FileWriter("test3.txt", true);  out.write("Hello4444444");  out.write("test");  out.write("abc444444");  out.write("\n");  out.write("xyz");  out.write("end444444444");  out.flush();  out.close();  System.out.println("done");  }  } | |
| |  | | --- | | A.  done |  |  | | --- | | B.  throws Exception |  |  | | --- | | C.  true | | | | |
|  | | | |

|  |  |
| --- | --- |
| **41.** | package io2;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  public class M4 {  public static void main(String[] args) {  File f1 = new File("test4.txt");  FileWriter out = null;  try  {  out = new FileWriter(f1);  out.write("hello");  out.write("test");  out.write("end");  }  catch(IOException ex)  {  ex.printStackTrace();  }  finally  {  if(out != null)  {  try  {  out.flush();  out.close();  }  catch(IOException ex)  {  ex.printStackTrace();  }  }  }    System.out.println("done");          }  }  //Whether it compiles successfull or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **42.** | package io2;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  public class M5 {  public static void main(String[] args) {  File f1 = new File("test5.txt");  try (FileWriter out = new FileWriter(f1)) //JDK1.7  {  out.write("hello");  out.write("test");  out.write("end");  }  catch(IOException ex)  {  ex.printStackTrace();  }  }  }  // Whether it compiles successfull are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **43.** | package io2;  import java.io.File;  import java.io.FileReader;  import java.io.IOException;  public class M6 {  public static void main(String[] args) throws IOException {  File f1 = new File("test3.txt");  char[] chars = new char[50];  FileReader fin = new FileReader(f1);  fin.read(chars);  fin.close();  String s1 = new String(chars);  System.out.println(s1);  }  }  // assume test3.txt fille in your system in that file we wrote some content by using the following comands  out.write("Hello4444444");  out.write("test");  out.write("abc444444");  out.write("\n");  out.write("xyz");  out.write("end444444444");  then what could be the output |
| |  | | --- | | A.  xyzend444444444Hello4444  Hello4444444testabc444444 |  |  | | --- | | B.  Hello4444444testabc444444  xyzend444444444Hello4444 |  |  | | --- | | C.  throws Exception | | | |
|  | | |
| **44.** | package io2;  import java.io.File;  import java.io.FileReader;  import java.io.IOException;  public class M7 {  public static void main(String[] args)  {  File f1 = new File("test3.txt");  FileReader fin = null;  char[] chars = new char[110];  try  {  fin = new FileReader(f1);  fin.read(chars);  }  catch(IOException ex)  {  ex.printStackTrace();  }  finally  {  if(fin != null)  {  try  {  fin.close();  }  catch(IOException ex)  {  ex.printStackTrace();  }  }  }  String s1 = new String(chars);  System.out.println(s1);    }  }  // assume test3.txt fille in your system in that file we wrote some content two times by using the following comands  out.write("Hello4444444");  out.write("test");  out.write("abc444444");  out.write("\n");  out.write("xyz");  out.write("end444444444");  then what could be the output |
| |  | | --- | | A.  xyzend444444444Hello4444  Hello4444444testabc444444 |  |  | | --- | | B.  Hello4444444testabc444444  xyzend444444444Hello4444 |  |  | | --- | | C.  Hello4444444testabc444444  xyzend444444444Hello4444444testabc444444  xyzend444444444 |  |  | | --- | | D.  Hello4444444testabc444444  xyzend444444444  Hello4444444testabc444444  xyzend444444444 | | | |
|  | | |

|  |  |
| --- | --- |
| **45.** | package io2;  import java.io.File;  import java.io.FileReader;  import java.io.IOException;  public class M8 {  public static void main(String[] args)  {  File f1 = new File("test3.txt");    try (FileReader fin = new FileReader(f1))  {  char[] chars = new char[110];  fin.read(chars);  String s1 = new String(chars);  System.out.println(s1);  }  catch(IOException ex)  {  ex.printStackTrace();  }  }  } |
| |  | | --- | | A.  Hello4444444testabc444444  xyzend444444444Hello4444444testabc444444  xyzend444444444 |  |  | | --- | | B.  Hello4444444testabc444444  xyzend444444444  Hello4444444testabc444444  xyzend444444444 |  |  | | --- | | C.  throws Exception |  |  | | --- | | D.  Compile Time Error | | | |
|  | | |
| **46.** | package io2;  import java.io.BufferedWriter;  import java.io.FileWriter;  import java.io.IOException;  public class M9 {  public static void main(String[] args) throws IOException {  FileWriter out = new FileWriter("test22.txt", true);  BufferedWriter bout = new BufferedWriter(out);  bout.write("test");  bout.write("test");  bout.newLine();  bout.write("test");  bout.write("test");  bout.newLine();  bout.write("test");  bout.write("test");  bout.newLine();  bout.write("test");  bout.write("test");  bout.flush();  bout.close();  out.close();  System.out.println("done");    }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  throws Exception | | | |
|  | | |

|  |  |
| --- | --- |
| **47.** | import java.io.Serializable;  public class B implements Serializable{  int i;  transient int j;  static int k;  B(int i, int j, int k)  {  this.i = i;  this.j = j;  this.k = k;  }  @Override  public String toString() {  return "(" + i + ", " + j + ", " + k + ")";  }  }  //whether it compiles successfully or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **48.** | import java.io.Serializable;  class C implements Serializable  {  int i;  }  public class D extends C{  int j;  D(int i, int j)  {  this.i = i;  this.j = j;  }  @Override  public String toString() {  return "(" + i + ", " + j + ")";  }  }  //whether it compiles successfully or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **49.** | import java.io.Serializable;  class E  {  int i;  E()  {  i = 1;  System.out.println("E()");  }  }  class F extends E  {  int j;  F()  {  j = 2;  System.out.println("F()");  }  }  public class G extends F implements Serializable{  int k;  G()  {  k = 3;  System.out.println("G()");  }  @Override  public String toString() {  return "(" + i + ", " + j + ", " + k + ")";  }  }  //whether it compiles successfully or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
|  | | | |
| **50.** | import java.io.IOException;  import java.io.ObjectInputStream;  import java.io.ObjectOutputStream;  import java.io.Serializable;  public class H implements Serializable  {  int x;  double y;  String z;    H(int x, double y, String z)  {  this.x = x;  this.y = y;  this.z = z;  }    @Override  public String toString() {  return "(" + x + ", " + y + ", " + z + ")";  }    private void writeObject(ObjectOutputStream out) throws IOException  {  System.out.println("i am from writeObject");  out.defaultWriteObject();  out.writeInt(5000);  }    private void readObject(ObjectInputStream in) throws IOException, ClassNotFoundException  {  System.out.println("i am from readObject");  in.defaultReadObject();  int data = in.readInt();  System.out.println(data);  }    }  //whether it compiles successfully or not? | |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
|  | | | |

|  |  |
| --- | --- |
| **51.** | import java.io.Externalizable;  import java.io.IOException;  import java.io.ObjectInput;  import java.io.ObjectInputStream;  import java.io.ObjectOutput;  import java.io.ObjectOutputStream;  import java.io.Serializable;  public class I implements Externalizable  {  int x;  double y;  String z;  I(int x, double y, String z)  {  this.x = x;  this.y = y;  this.z = z;  }  @Override  public String toString() {  return "(" + x + ", " + y + ", " + z + ")";  }  @Override  public void writeExternal(ObjectOutput out) throws IOException {  out.writeInt(x);  //out.writeDouble(y);  out.writeUTF(z);  }    @Override  public void readExternal(ObjectInput in) throws IOException, ClassNotFoundException {  x = in.readInt();  //y = in.readDouble();  z = in.readUTF();  }  } |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

Bottom of Form