|  |  |
| --- | --- |
| **1.** | class M1  {  public static void main(String[] args)  {  String s1 = "hello";  System.out.println(s1);  String s2 = new String("hello");  System.out.println(s2);  }  } |
| |  | | --- | | A.  hello  hello |  |  | | --- | | B.  hello |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  none | | | |
| **Correct Answer: A** | | |
| **2.** | class M2  {  public static void main(String[] args)  {  String s1 = "hello";  String s2 = "hello";  System.out.println(s1 == s2);  }  } |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **3.** | class M3  {  public static void main(String[] args)  {  String s1 = new String("hello");  String s2 = new String("hello");  System.out.println(s1 == s2);  }  } |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **4.** | class M4  {  public static void main(String[] args)  {  String s1 = "hello";  String s2 = "Hello";  String s3 = new String("hello");  String s4 = new String("hello");  System.out.println(s3 .equals(s4));  }  } |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **5.** | class M5  {  public static void main(String[] args)  {  String s1 = "hello";  int i = s1.length();  System.out.println(i);  System.out.println(s1.length());  }  } |
| |  | | --- | | A.  5  5 |  |  | | --- | | B.  4  4 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **6.** | class M6  {  public static void main(String[] args)  {  String s1 = "a c";  String s2 = " a c ";  System.out.println(s1.length());  System.out.println(s2.length());  }  } |
| |  | | --- | | A.  5  3 |  |  | | --- | | B.  3  5 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **7.** | class M7  {  public static void main(String[] args)  {  String s1 = "Hi India";  String s2 = "Hi "India";  System.out.println(s1);  System.out.println(s2);  }  } |
| |  | | --- | | A.  Hi India  Hi "India |  |  | | --- | | B.  Hi India  Hi India |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: C** | | |
| **8.** | class M8  {  public static void main(String[] args)  {  String s1 = "Hi India";  String s2 = "Hi \"India\"";  System.out.println(s1.length());  System.out.println(s2.length());  }  } |
| |  | | --- | | A.  10  8 |  |  | | --- | | B.  8  10 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |

|  |  |  |
| --- | --- | --- |
| **9.** | class M9  {  public static void main(String[] args)  {  String s1 = "hello n hello";  String s2 = "hello \n hello";  System.out.println(s1);  System.out.println(s2);  }  } | |
| |  | | --- | | A.  hello n hello  hello  hello |  |  | | --- | | B.  hello n hello  hello |  |  | | --- | | C.  Compilation error | | | | |
| **Correct Answer: A** | | | |
| **10.** | | class M10  {  public static void main(String[] args)  {  String s1 = "abc \ xyz";  System.out.println(s1);  }  }  // Will it compiles fine or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | | |
| **Correct Answer: B** | | | |

|  |  |
| --- | --- |
| **11.** | class M11  {  public static void main(String[] args)  {  String s1 = "hello \\ xyz";  System.out.println(s1);  }  }  // will it com[piles fine or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
| **Correct Answer: A** | | |
| **12.** | class M12  {  public static void main(String[] args)  {  String path = "D:\JDK8.0\bin";  System.out.println(path);  }  } |
| |  | | --- | | A.  D:\JDK8.0\bin |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **13.** | class M13  {  public static void main(String[] args)  {  String path = "D:\\JDK8.0\\bin";  System.out.println(path);  }  } |
| |  | | --- | | A.  D:\JDK8.0\bin |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **14.** | class M14  {  public static void main(String[] args)  {  String path = "D:/JDK8.0/bin";  System.out.println(path);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  D:/JDK8.0/bin | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **15.** | class M15  {  public static void main(String[] args)  {  String s1 = "abc";  s1 = s1 + "xyz";  System.out.println(s1);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  abcxyz |  |  | | --- | | D.  abc + xyz |  |  | | --- | | E.  None | | | |
| **Correct Answer: C** | | |
| **16.** | class M16  {  public static void main(String[] args)  {  String s1 = "abc";  String s2 = null;  String s3 = s1 + s2;  String s4 = s2 + s1;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  System.out.println(s4);  }  } |
| |  | | --- | | A.  abc  null  abcnull  nullabc |  |  | | --- | | B.  abc  null  abcnull  abcnull |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  null  abc  abcnull  nullabc | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **17.** | class M17  {  public static void main(String[] args)  {  String s1 = null;  String s2 = s1 + null;  String s3 = null + s1;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  }  } |
| |  | | --- | | A.  null  null  nullnull |  |  | | --- | | B.  null  nullnull  nullnull |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **18.** | class M18  {  public static void main(String[] args)  {  String s1 = null + null;  System.out.println(s1);  }  } |
| |  | | --- | | A.  nullnull |  |  | | --- | | B.  null + null |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **19.** | class M19  {  public static void main(String[] args)  {  String s1 = "abc" + 5 + 6;  String s2 = 5 + "abc" + 6;  String s3 = 5 + 6 + "abc";  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  }  } |
| |  | | --- | | A.  abc56  5abc6  11abc |  |  | | --- | | B.  abc11  5abc6  11abc |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **20.** | class M20  {  public static void main(String[] args)  {  String s1 = null + null;  String s2 = 5 + 9;  String s3 = null + "";  String s4 = "" + null;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  System.out.println(s4);  }  } |
| |  | | --- | | A.  nullnull  14  null  null |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  runtime error | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **21.** | class M21  {  public static void main(String[] args)  {  String s1 = null;  System.out.println(s1.length());  }  }  // will it compiles fine or not? |
| |  | | --- | | A.  yes | | | |
| **Correct Answer: A** | | |
| **22.** | class M22  {  public static void main(String[] args)  {  String s1 = NULL;  System.out.println("Hello World!"s1.length();  }  }  //will it compiles fine or not? |
| |  | | --- | | A.  yes |  |  | | --- | | B.  No | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **23.** | .class M23  {  public static void main(String[] args)  {  String s1 = "null";  System.out.println(s1.length());  }  } |
| |  | | --- | | A.  4 |  |  | | --- | | B.  NullPointerException |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **24.** | class M24  {  public static void main(String[] args)  {  String s1 = "xyz";  System.out.println(s1);  s1.concat("hello");  System.out.println(s1);  }  } |
| |  | | --- | | A.  xyz  xyz |  |  | | --- | | B.  xyzxyz |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **25.** | class M25  {  public static void main(String[] args)  {  String s1 = "xyz";  System.out.println(s1);  String s2 = s1.concat("hello");  System.out.println(s1);  System.out.println(s2);  }  } |
| |  | | --- | | A.  xyzxyz  xyzhello |  |  | | --- | | B.  xyz  xyz  xyz  hello |  |  | | --- | | C.  xyz  xyz  xyzhello |  |  | | --- | | D.  Compilation error | | | |
| **Correct Answer: C** | | |
| **26.** | package StringBuffer;  public class S1 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer();  sb.append("hello");  sb.append("test");  sb.append("123");  System.out.println(sb);  }  } |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  false  false |  |  | | --- | | C.  true  false |  |  | | --- | | D.  false  true | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **27.** | package StringBuffer;  public class S13 {  public static void main(String[] args) {  StringBuilder sb = new StringBuilder("hellotoall");  System.out.println(sb);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  123 |  |  | | --- | | C.  hellotest123 | | | |
| **Correct Answer: C** | | |
| **28.** | package StringBuffer;  public class S3 {  public static void main(String[] args) {  StringBuffer sb1 = new StringBuffer();  sb1.append("hello");    System.out.println(sb1.length());  System.out.println(sb1.capacity());  }  } |
| |  | | --- | | A.  5  16 |  |  | | --- | | B.  5  34 |  |  | | --- | | C.  5  32 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **29.** | package StringBuffer;  public class S4 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer();  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  }  } |
| |  | | --- | | A.  16  1  ---------------------  18  34 |  |  | | --- | | B.  16  0  ---------------------  18  32 |  |  | | --- | | C.  18  34  ---------------------  16  0 |  |  | | --- | | C.  16  0  ---------------------  18  34 | | | |
| **Correct Answer: C** | | |
| **30.** | package StringBuffer;  public class S5 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer(2000);  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  }  } |
| |  | | --- | | A.  2000  0  ---------------------  18  2000 |  |  | | --- | | B.  2000  1  ---------------------  18  2000 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **31.** | package StringBuffer;  public class S6 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  System.out.println("---------------------");  sb.trimToSize();  System.out.println(sb.length());  System.out.println(sb.capacity());  }  } |
| |  | | --- | | A.  16  4  ---------------------  22  44  ---------------------  22  22 |  |  | | --- | | B.  20  4  ---------------------  22  42  ---------------------  22  22 |  |  | | --- | | C.  20  4  ---------------------  22  42  ---------------------  22  44 | | | |
| **Correct Answer: B** | | |
| **32.** | package StringBuffer;  public class S7 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  System.out.println("---------------------");  sb.trimToSize();  System.out.println(sb.length());  System.out.println(sb.capacity());  }  } |
| |  | | --- | | A.  20  4  ---------------------  22  42  ---------------------  22  22 |  |  | | --- | | B.  16  4  ---------------------  22  44  ---------------------  22  22 |  |  | | --- | | C.  20  4  ---------------------  22  42  ---------------------  22  44 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **33.** | package StringBuffer;  public class S8 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb);  sb.reverse();  System.out.println(sb);  }  } |
| |  | | --- | | A.  tset |  |  | | --- | | B.  test  tset |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **34.** | package StringBuffer;  public class S9 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb);  sb.deleteCharAt(2);  System.out.println(sb);  }  } |
| |  | | --- | | A.  test  tet |  |  | | --- | | B.  tet |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **35.** | package StringBuffer;  public class S10 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(2, 6);  System.out.println(sb);  }  } |
| |  | | --- | | A.  hellotoall  heloall |  |  | | --- | | B.  hellotoall  hetoall |  |  | | --- | | C.  hellotoall  heoall |  |  | | --- | | D.  Compilation error | | | |
| **Correct Answer: C** | | |
| **36.** | package StringBuffer;  public class S11 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(2, 30);  System.out.println(sb);  }  } |
| |  | | --- | | A.  hellotoall  hel |  |  | | --- | | B.  StringIndexOutOfBoundsException |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  hellotoall  he | | | |
| **Correct Answer: D** | | |

|  |  |
| --- | --- |
| **37.** | package StringBuffer;  public class S12 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(20, 30);  System.out.println(sb);  }  } |
| |  | | --- | | A.  hellotoall  StringIndexOutOfBoundsException |  |  | | --- | | B.  StringIndexOutOfBoundsException |  |  | | --- | | C.  hellotoall | | | |
| **Correct Answer: A** | | |
| **38.** | package StringBuffer;  public class S13 {  public static void main(String[] args) {  StringBuilder sb = new StringBuilder("hellotoall");  System.out.println(sb);  }  } |
| |  | | --- | | A.  hellotoall |  |  | | --- | | B.  No output |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  StringOutOfBoundsException | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **39.** | package formatter;  public class S1 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format("==> %s", s1);  System.out.println(s1);  System.out.println(s2);  }  } |
| |  | | --- | | A.  hello==> hello <== |  |  | | --- | | B.  hello  ==> hello |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **40.** | package formatter;  public class S2 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format("==> %s <==", s1);  System.out.println(s1);  System.out.println(s2);  }  } |
| |  | | --- | | A.  hello  ==> hello <== |  |  | | --- | | B.  hello==> hello <== |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **41.** | package formatter;  public class S3 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format(" (%s) and (%d) ans (%f)", s1, 100, 4.5);  System.out.println(s1);  System.out.println(s2);  }  } |
| |  | | --- | | A.  hello  (hello) and (100) ans (4.500000) |  |  | | --- | | B.  hello (hello) and (100) ans (4.500000) |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **42.** | package formatter;  public class S4 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %d ", i);  System.out.printf("j value is %f ", d);  System.out.printf("b value is %b ", b);  System.out.printf("s1 value is %s ", s1);  System.out.printf("ch value is %c ", ch);  }  } |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **43.** | package formatter;  public class S5 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %d ", i);  System.out.printf("\nj value is %f ", d);  System.out.printf("\nb value is %b ", b);  System.out.printf("\ns1 value is %s ", s1);  System.out.printf("\nch value is %c ", ch);  }  } |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |
| **44.** | package formatter;  public class S6 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %s ", i);  System.out.printf("\nj value is %s ", d);  System.out.printf("\nb value is %s ", b);  System.out.printf("\ns1 value is %s ", s1);  System.out.printf("\nch value is %s ", ch);  }  } |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **45.** | package formatter;  public class S7 {  public static void main(String[] args) {  double d = 2.5;  System.out.printf("\nj value is %d ", d);    }  }  // Will it compiles fine or not |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
| **Correct Answer: B** | | |
| **46.** | package formatter;  public class S8 {  public static void main(String[] args) {  int i = 100;  double j = 2.5;  System.out.printf("i value is %d and j value is %f",i, j);    }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  i value is 100 and d value is 2.500000 |  |  | | --- | | C.  i value is 100 and d value is 2.5 | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **47.** | package formatter;  public class S9 {  public static void main(String[] args) {  int i = 100;  double j = 2.5;  System.out.printf("j value is %2$f and j value is %1$d",i, j);    }  } |
| |  | | --- | | A.  j value is 2.500000 and j value is 100 |  |  | | --- | | B.  j value is 2.5 and j value is 100 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |
| **48.** | package formatter;  public class S10 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%d)", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  123456 |  |  | | --- | | B.  (123456) |  |  | | --- | | C.  num is (123456) |  |  | | --- | | D.  Compilation error | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **49.** | package formatter;  public class S11 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%020d)", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  num is (0000000000000,123456) |  |  | | --- | | B.  num is (0000000000000123456) |  |  | | --- | | C.  num is (0000000000000123,456) |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **50.** | package formatter;  public class S12 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%-20d)", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  num is (123456) |  |  | | --- | | B.  num is (123456 ) |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **51.** | package formatter;  public class S13 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%-020d)", num);  System.out.println(s1);  }  }  // Will it compiles fine or not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
| **Correct Answer: B** | | |
| **52.** | package formatter;  public class S14 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%0,20d)", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  num is (0000000000000,123456) |  |  | | --- | | B.  num is (0000000000000123456) |  |  | | --- | | C.  num is (0000000000000123,456) |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **53.** | package formatter;  public class S15 {  public static void main(String[] args) {  int num = -123456;  String s1 = String.format("num is %(d", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  num is (123456) |  |  | | --- | | C.  123456 | | | |
| **Correct Answer: B** | | |
| **54.** | package formatter;  public class S16 {  public static void main(String[] args) {  double num = 1234.5678;  String s1 = String.format("num is %.3f", num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  num is 1234.568 |  |  | | --- | | B.  num is 1234.567 |  |  | | --- | | C.  num is 12345.678 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **55.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S1 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "and";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  6:and  16:and  32:and |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  and  and  and |  |  | | --- | | D.  6:  16:  32: | | | |
| **Correct Answer: A** | | |
| **56.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S2 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "hello";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  hello  hello  hello |  |  | | --- | | B.  0:hello  10:hello  20:hello |  |  | | --- | | C.  0:  10:  20: |  |  | | --- | | D.  Compilation error | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **57.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S3 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "a";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  6:a  16:a  28:a  32:a  38:a |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  6:  16:  28:  32:  38: | | | |
| **Correct Answer: A** | | |
| **58.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "[ali]";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  No output |  |  | | --- | | C.  2:l  3:l  6:a  12:l  13:l  16:a  22:l  23:l  28:a  29:i  32:a  38:a  39:i | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **59.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\s";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  6:  14:  15:  16:  17:  18:  26:  27:  28:  29:  30:  31:  32:  33:  41:  42:  43:  44:  45:  46:  47:  48:  49: |  |  | | --- | | B.  6:  14:  26:  41: |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |
| **60.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\s+";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  6:  14:  15:  16:  17:  18:  26:  27:  28:  29:  30:  31:  32:  33:  41:  42:  43:  44:  45:  46:  47:  48:  49: |  |  | | --- | | B.  6:  14:  26:  41: |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **61.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S5{  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\d";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  4:8  11:10  23:11  38:12  56:18 |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  4:8  11:1  12:0  23:1  24:1  38:1  39:2  56:1  57:8 | | | |
| **Correct Answer: C** | | |
| **62.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S5{  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\d+";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  4:8  11:10  23:11  38:12  56:18 |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  4:8  11:1  12:0  23:1  24:1  38:1  39:2  56:1  57:8 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **63.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S6 {  public static void main(String[] args) {  String src = "java@java.com";  String regex = "@";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  4:@ |  |  | | --- | | C.  4: |  |  | | --- | | D.  @ | | | |
| **Correct Answer: B** | | |
| **64.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S6 {  public static void main(String[] args) {  String src = "java@java.com";  String regex = ".";  //String regex = "@";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  System.out.println(matches.find());  }  } |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **65.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  No output | | | |
| **Correct Answer: A** | | |
| **66.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(0);  System.out.println(date);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  No output | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **67.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(1000\*60\*24);  System.out.println(date);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | | |
| **Correct Answer: C** | | |
| **68.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(-1000\*60\*60\*24);  System.out.println(date);  }  } |
| |  | | --- | | A.  Wed Dec 31 05:30:00 IST 1969 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **69.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Tue Dec 17 14:14:38 IST 2019 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | | |
| **Correct Answer: B** | | |
| **70.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.DATE, 20);  Date date = cal.getTime();  System.out.println(date);  }  } |
| |  | | --- | | A.  Mon Jan 06 14:18:12 IST 2020 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **71.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.MONTH, -5);  Date date = cal.getTime();  System.out.println(date);  }  } |
| |  | | --- | | A.  Mon Jan 06 14:18:12 IST 2020 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Wed Jul 17 14:20:28 IST 2019 | | | |
| **Correct Answer: C** | | |
| **72.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.YEAR, 10);  Date date = cal.getTime();  System.out.println(date);  }  } |
| |  | | --- | | A.  Mon Dec 17 14:21:55 IST 2029 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Wed Jul 17 14:20:28 IST 2019 | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **73.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getInstance();  String s1 = df.format(date);  System.out.println(s1);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 AM |  |  | | --- | | B.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 PM |  |  | | --- | | C.  Tue Dec 17 14:24:18 IST 2020  12/17/19 2:24 PM | | | |
| **Correct Answer: B** | | |
| **74.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);  String s1 = df.format(date);  System.out.println(s1);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 AM |  |  | | --- | | B.  Tue Dec 17 14:27:13 IST 2019  12/17/19 |  |  | | --- | | B.  Tue Dec 17 14:27:13 IST 2019  12/17/20 | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **75.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {    Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.LONG);  String s1 = df.format(date);  System.out.println(s1);  }  } |
| |  | | --- | | A.  Tue Dec 17 14:28:33 IST 2019  December 17, 2019 |  |  | | --- | | B.  Mon 14:28:33 IST 2020  December 17, 2020 |  |  | | --- | | C.  Tue Dec 17 14:28:33 IST 2018  December 17, 2018 | | | |
| **Correct Answer: A** | | |
| **76.** | package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);  String s1 = df.format(date);  System.out.println(s1);    try {  Date date2 = df.parse(s1);  System.out.println(date2);  }  catch(ParseException ex) {  ex.printStackTrace();  }  }  } |
| |  | | --- | | A.  Tue Dec 17 14:32:08 IST 2019  Tuesday, December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 14:32:08 IST 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **77.** | package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  SimpleDateFormat sdf = new SimpleDateFormat("dd----MM----YY");  String s1 = sdf.format(date);  System.out.println(s1);  }  } |
| |  | | --- | | B.  Tue Dec 17 14:37:40 IST 2019  17----Dec----2019 |  |  | | --- | | C.  Tue Dec 17 14:37:40  17----Dec----2019 | | | |
| **Correct Answer: A** | | |
| **78.** | "package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  SimpleDateFormat sdf = new SimpleDateFormat(""dd----MMM----YYYY"");  String s1 = sdf.format(date);  System.out.println(s1);  }  }  " |
| |  | | --- | | B.  Tue Dec 17 14:37:40 IST 2019  17----Dec----2019 |  |  | | --- | | C.  Tue Dec 17 14:37:40  17----Dec----2019 | | | |
| **Correct Answer: B** | | |

|  |  |
| --- | --- |
| **79.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  double num = 4.56787887;  System.out.println(num);  NumberFormat nf = NumberFormat.getInstance();  String s1 = nf.format(num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  4.56787887  4.568 |  |  | | --- | | B.  4.56787887  4.56877 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |
| **80.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S7 {  public static void main(String[] args) {  double num = 423654778954.56787887;  System.out.println(num);  NumberFormat nf = NumberFormat.getCurrencyInstance(Locale.UK);  String s1 = nf.format(num);  System.out.println(s1);  }  } |
| |  | | --- | | A.  4.236547789545679E11  $423,654,778,954.57 |  |  | | --- | | B.  4.236547789545679E11  £423,654,778,954.57 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: B** | | |

|  |  |  |
| --- | --- | --- |
| **81.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL, Locale.FRANCE);  String s1 = df.format(date);  System.out.println(s1);  }  } | |
| |  | | --- | | A.  Tue Dec 17 14:49:54 IST 2019  mardi 17 décembre 2019 |  |  | | --- | | B.  Tue Dec 17 14:49:54 IST 2019  17 décembre 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | | |
| **Correct Answer: A** | | | |
| **82.** | package dateAndNumberFormats;  import java.util.Date;  public class S1 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  }  }  // what is the output date |
| |  | | --- | | A.  Today's date |  |  | | --- | | B.  Tomorrow's date |  |  | | --- | | C.  Yesterday's date |  |  | | --- | | D.  None | | | | |
| **Correct Answer: A** | | | |

|  |  |
| --- | --- |
| **83.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S2 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance();  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 15:24:16 IST 2019  Dec 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:24:16 IST 2019  17-Dec-2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |
| **84.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S3 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 16:28:12 IST 2019  12/17/2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:28:12 IST 2019  Dec/17/19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 16:28:12 IST 2019  12/17/19  Tue Dec 17 00:00:00 IST 2019 | | | |
| **Correct Answer: C** | | |

|  |  |
| --- | --- |
| **85.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S4 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.MEDIUM);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 16:24:52 IST 2019  December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:27:28 IST 2019  Dec 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: B** | | |
| **86.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S5 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.LONG);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 16:24:52 IST 2019  December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:24:52 IST 2019  17, December, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **87.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S6 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 16:23:37 IST 2019  December 17, 2019, Tuesday  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:23:37 IST 2019  Tuesday, December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error | | | |
| **Correct Answer: B** | | |
| **88.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class S7 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("dd-MM-yyyy");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 15:35:54 IST 2019  17-12-2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:35:54 IST 2019  17-Dec--2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 15:35:54 IST 2019  17-Dec--19Tue Dec 17 00:00:00 IST 2019 | | | |
| **Correct Answer: A** | | |

|  |  |  |
| --- | --- | --- |
| **89.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class S8 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyy.MM.dd G 'at' HH:mm:ss z");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } | |
| |  | | --- | | A.  Tue Dec 17 15:41:03 IST 2019  17.12.2019 AD at 15:41:03 IST  Tue Dec 17 15:41:03 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:41:03 IST 2019  2019.12.17 AD at 15:41:03 IST  Tue Dec 17 15:41:03 IST 2019 |  |  | | --- | | C.  Compilation error | | | | |
| **Correct Answer: B** | | | |
| **90.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S9 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("h:mm a");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 15:45:07 IST 2019  3:45:03 PM  Thu Jan 01 15:45:00 IST 1970 |  |  | | --- | | B.  Tue Dec 17 15:45:07 IST 2019  3:45 PM  Thu Jan 01 15:45:00 IST 1970 |  |  | | --- | | C.  Compilation error | | | | |
| **Correct Answer: B** | | | |

|  |  |
| --- | --- |
| **91.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S10 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  calci.add(Calendar.DATE, 5);  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyyy.MMMMM.dd GGG hh:mm aaa");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Sun Dec 22 15:49:52 IST 2019  2019.December.22 AD 03:49 PM  Sun Dec 22 15:49:00 IST 2019 |  |  | | --- | | C.  Sun Dec 22 15:49:52 IST 2019  02019.December.22 AD 03:49 PM  Sun Dec 22 15:49:00 IST 2019 | | | | |
| **Correct Answer: C** | | | |
| **92.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S11 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  calci.add(Calendar.MONTH, -5);  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyyy.MM.dd'T'HH:mm:ss.SSSXXX");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } | |
| |  | | --- | | A.  Wed Jul 17 15:55:24 IST 2019  02019.07.17T15:55:24.874+05:30  Wed Jul 17 15:55:24 IST 2019 |  |  | | --- | | B.  Wed Jul 17 15:55:24 IST 2019  02019.07.17T15:55:24.874+05:30:45  Wed Jul 17 15:55:24 IST 2019 |  |  | | --- | | C.  Compilation error | | | | |
| **Correct Answer: A** | | | |

|  |  |
| --- | --- |
| **93.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  public class S12 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getInstance();  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  } |
| |  | | --- | | A.  385897.88888888  385,897.889  385,897.889 |  |  | | --- | | B.  385897.88888888  385,897.889  385897.889 |  |  | | --- | | C.  Compilation Error | | | |
| **Correct Answer: B** | | |
| **94.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  public class S13 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getCurrencyInstance();  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  } |
| |  | | --- | | A.  385897.88888888  $385,897.89  385897.89 |  |  | | --- | | B.  385897.88888888  385 897,89 ¤  385897.89 |  |  | | --- | | C.  385897.88888888  £385,897.89  385897.89 |  |  | | --- | | D.  None | | | |
| **Correct Answer: A** | | |

|  |  |
| --- | --- |
| **95.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S14 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("EEEE MMMM dd yy", Locale.FRANCE);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  } |
| |  | | --- | | A.  Tue Dec 17 16:06:51 IST 2019  Tuesday December 17 19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:06:12 IST 2019  mardi décembre 17 19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 16:08:25 IST 2019  martedì dicembre 17 19  Tue Dec 17 00:00:00 IST 2019 | | | |
| **Correct Answer: B** | | |
| **96.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  import java.util.Locale;  public class S15 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getInstance(Locale.ITALY);  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  } |
| |  | | --- | | A.  385897.88888888  385,897.89  385,897.89 |  |  | | --- | | B.  385897.88888888  385,897.89  385897.89 |  |  | | --- | | C.  385897.88888888  385.897,889  385897.889 | | | |
| **Correct Answer: C** | | |

Bottom of Form