|  |  |
| --- | --- |
| **1.** | class M1  {  public static void main(String[] args)  {  int i = 10;  Integer obj1 = new Integer(i);  Integer obj2 = Integer.valueOf(i);  int j = obj1.intValue();  int k = obj2.intValue();  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  Compilation Error | | | |
|  | | |
| **2.** | class M2  {  public static void main(String[] args)  {  Double obj1 = new Double(5.5);  Double obj2 = Double.valueOf(5.5);  double d1 = obj1.doubleValue();  double d2 = obj2.doubleValue();  System.out.println("done");  }  }  Will it compiles fine are Not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **3.** | class M3  {  public static void main(String[] args)  {  Character obj1 = new Character('a');  Character obj2 = Character.valueOf('a');  char c1 = obj1.charValue();  char c2 = obj2.charValue();  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **4.** | class M4  {  public static void main(String[] args)  {  boolean flag = false;  Boolean b1 = new Boolean(flag);  Boolean b2 = Boolean.valueOf(flag);  boolean i = b1.booleanValue();  boolean j = b2.booleanValue();  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **5.** | class M5  {  public static void main(String[] args)  {  String s1 = "10";  Integer obj1 = new Integer(s1);  Integer obj2 = Integer.valueOf(s1);  int i = obj1.intValue();  int j = obj2.intValue();  System.out.println(i);  System.out.println(j);  }  } |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  10  10 |  |  | | --- | | C.  0  0 | | | |
|  | | |
| **6.** | class M6  {  public static void main(String[] args)  {  String s1 = "10.5";  Double obj1 = new Double(s1);  Double obj2 = Double.valueOf(s1);  double d1 = obj1.doubleValue();  double d2 = obj2.doubleValue();  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **7.** | class M7  {  public static void main(String[] args)  {  String s1 = "true";  Boolean obj1 = new Boolean(s1);  Boolean obj2 = Boolean.valueOf(s1);  boolean b1 = obj1.booleanValue();  boolean b2 = obj2.booleanValue();  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **8.** | class M8  {  public static void main(String[] args)  {  String s1 = "10";  int i = Integer.parseInt(s1);  double j = Double.parseDouble(s1);  System.out.println(i);  System.out.println(j);  }  } |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  10  10 |  |  | | --- | | C.  10  10.0 | | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **9.** | class M9  {  public static void main(String[] args)  {  int i = 10;  double j = 1.5;  String s1 = Integer.toString(i);  String s2 = Double.toString(j);  System.out.println(s1);  System.out.println(s2);  }  } | |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  10  1.5 |  |  | | --- | | C.  10  15.0 | | | | |
|  | | | |
| **10.** | | class M10  {  public static void main(String[] args)  {  String s1 = "10r";  int i = Integer.parseInt(s1);  System.out.println("done");  }  } |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  NumberFormatException |  |  | | --- | | C.  10r  done |  |  | | --- | | D.  None | | | | |
|  | | | |

|  |  |
| --- | --- |
| **11.** | class M11  {  public static void main(String[] args)  {  String s1 = "10.5";  int i = Integer.parseInt(s1);  System.out.println("done");  }  } |
| |  | | --- | | A.  NumberFormatException |  |  | | --- | | B.  10.5  done |  |  | | --- | | C.  CompilationError |  |  | | --- | | D.  None | | | |
|  | | |
| **12.** | class M12  {  public static void main(String[] args)  {  String s1 = "10.5";  double i = Double.parseDouble(s1);  System.out.println("done");  }  } |
| |  | | --- | | A.  NumberFormatException |  |  | | --- | | B.  10.5  done |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  done | | | |
|  | | |

|  |  |
| --- | --- |
| **13.** | class M13  {  public static void main(String[] args)  {  String s1 = "xyz";  boolean flag = Boolean.parseBoolean(s1);  System.out.println(flag);  }  } |
| |  | | --- | | A.  NumberFormatException |  |  | | --- | | B.  false |  |  | | --- | | C.  true |  |  | | --- | | D.  Compilation Error | | | |
|  | | |
| **14.** | class M14  {  public static void main(String[] args)  {  Integer obj = new Integer(90);  int i = obj;  System.out.println("done");  }  } |
| |  | | --- | | A.  done |  |  | | --- | | B.  Compilation Error | | | |
|  | | |

|  |  |
| --- | --- |
| **15.** | class M15  {  public static void main(String[] args)  {  int i = 10;  Integer obj1 = i;  Integer obj2 = 20;  System.out.println("done");  }  }  Will it compilation fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **16.** | class M16  {  static void test(int i)  {  System.out.println("test(int)");  }  public static void main(String[] args)  {  test(10);  System.out.println("------------");  Integer obj = new Integer(90);  test(obj);  }  } |
| |  | | --- | | A.  test(int)  ------------  test(int) |  |  | | --- | | B.  test(int)  test(int)  ------------ |  |  | | --- | | C.  ------------  test(int)  test(int) |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **17.** | class M17  {  static void test(Integer i)  {  System.out.println("test(Integer)");  }  public static void main(String[] args)  {  test(10);  System.out.println("------------");  Integer obj = new Integer(90);  test(obj);  }  } |
| |  | | --- | | A.  test(Integer)  test(Integer)  ------------ |  |  | | --- | | B.  ------------  test(Integer)  test(Integer) |  |  | | --- | | C.  test(Integer)  ------------  test(Integer) | | | |
|  | | |
| **18.** | class M18  {  static void test(double j)  {  System.out.println("test(double)");  }  public static void main(String[] args)  {  test(10);  System.out.println("done");  int i = 200;  test(i);  }  } |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  test(double)  done  test(double) |  |  | | --- | | C.  test(double)  test(double)  done | | | |
|  | | |

|  |  |
| --- | --- |
| **19.** | class M19  {  public static void main(String[] args)  {  int i = 10;  double j = 10.5;  j = i;  Double obj1 = new Double(20);  Integer obj2 = new Integer(20);  obj1 = obj2;  System.out.println("done");  }  } |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  NumberFormatException |  |  | | --- | | C.  done | | | |
|  | | |
| **20.** | class M20  {  public static void main(String[] args)  {  Double obj1 = new Double(20);  Integer obj2 = new Integer(20);  Number obj3 = obj1;  Number obj4 = obj2;  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **21.** | class M21  {  static void test1(Integer obj)  {  System.out.println("test1(Integer)");  }  public static void main(String[] args)  {  byte b1 = 10;  test1(b1);  System.out.println("done");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **22.** | class M22  {  static void test1(Integer obj)  {  System.out.println("test1(Integer)");  }  static void test1(Byte obj)  {  System.out.println("test1(Byte)");  }  public static void main(String[] args)  {  byte b1 = 10;  test1(b1);  System.out.println("done");  }  } |
| |  | | --- | | A.  test1(Byte)  done |  |  | | --- | | B.  done  test1(Byte) |  |  | | --- | | C.  Compile Time Error | | | |
|  | | |

|  |  |
| --- | --- |
| **23.** | class M23  {  static void test1(Integer obj)  {  System.out.println("test1(Integer)");  }  static void test1(Byte obj)  {  System.out.println("test1(Byte)");  }  static void test1(double d1)  {  System.out.println("test1(double)");  }  public static void main(String[] args)  {  byte b1 = 10;  test1(b1);  System.out.println("done");  }  } |
| |  | | --- | | A.  done  test1(double) |  |  | | --- | | B.  test1(double)  done |  |  | | --- | | C.  Compile Time Error | | | |
|  | | |
| **24.** | class M24  {  static void test1(Integer obj)  {  System.out.println("test1(Integer)");  }  static void test1(Number obj)  {  System.out.println("test1(Number)");  }  public static void main(String[] args)  {  byte b1 = 10;  test1(b1);  System.out.println("done");  }  } |
| |  | | --- | | A.  test1(Number)  done |  |  | | --- | | B.  test1(Integer)  done |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **25.** | class M25  {  static void test1(Integer obj)  {  System.out.println("test1(Integer)");  }  static void test1(Object obj)  {  System.out.println("test1(Object)");  }  public static void main(String[] args)  {  byte b1 = 10;  test1(b1);  System.out.println("done");  }  } |
| |  | | --- | | A.  test1(Integer)  done |  |  | | --- | | B.  test1(Object)  done |  |  | | --- | | C.  done |  |  | | --- | | D.  Compilation Error | | | |
|  | | |
| **26.** | class M26  {  static void test(int ... args)  {  System.out.println("var arg");  }  public static void main(String[] args)  {  test();  test(1);  test(2, 4);  test(40, 10);  System.out.println("done");  }  } |
|  | | |
| **, ,** | | |

|  |  |
| --- | --- |
| **27.** | class M27  {  static void test(int ... args)  {  System.out.println("var arg:" + args.length);  }  public static void main(String[] args)  {  test();  test(1);  test(2, 4);  test(40, 10, 5);  test(40, 10, 5, 45, 100);  System.out.println("done");  }  } |
| |  | | --- | | A.  var arg:1  var arg:2  var arg:4  var arg:5  done |  |  | | --- | | B.  var arg  var arg  var arg  done |  |  | | --- | | C.  var arg:  var arg:  var arg:  var arg:  done |  |  | | --- | | D.  Compilation Error | | | |
|  | | |
| **28.** | class M28  {  static void test(int ... args)  {  for(int value : args)  {  System.out.println(value);  }  System.out.println("------------");  }  public static void main(String[] args)  {  test();  test(1);  test(2, 4);  test(40, 10, 5);  test(40, 10, 5, 45, 100);  System.out.println("done");  }  } |
| |  | | --- | | A.  ------------  1  ------------  2  4  ------------  40  10  5  ------------  40  10  5  45  100  ------------  done |  |  | | --- | | B.  ------------  1  2  4  40  10  5  40  10  5  45  100  ------------  done |  |  | | --- | | C.  Compilation Error | | | |
|  | | |

|  |  |
| --- | --- |
| **29.** | class M29  {  public static void main(String ... args)  {  System.out.println("Hello World!");  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **30.** | class M30  {  public static void test(String x, double ... y)  {  System.out.println(x);  for(double value : y)  {  System.out.println(value);  }  System.out.println("--------");  }  public static void main(String[] args)  {  test("abc");  test("abc", 4.5);  test("abc", 4.5, 50.0);  test("abc", 4.5, 50.0, 2.3);  System.out.println("done");  }  } |
| |  | | --- | | A.  --------  abc  --------  abc  4.5  --------  abc  4.5  50.0  --------  abc  4.5  50.0  2.3  done |  |  | | --- | | B.  abc  --------  abc  4.5  --------  abc  4.5  50.0  --------  abc  4.5  50.0  2.3  --------  done |  |  | | --- | | C.  --------  abc  4.5  --------  abc  4.5  50.0  --------  abc  4.5  50.0  2.3  --------  done |  |  | | --- | | D.  Compilation Error | | | |
|  | | |

|  |  |
| --- | --- |
| **31.** | class M31  {  public static void test(double ... y, String x)  {  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **32.** | class M32  {  public static void test(double ... y)  {  }  public static void test(double[] z)  {  }  }  Will it compiles fine are not? |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **33.** | class M34  {  public static void test(byte b1)  {  System.out.println("byte");  }  public static void test(double b1)  {  System.out.println("double");  }  public static void test(Byte b1)  {  System.out.println("Byte");  }  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(Number b1)  {  System.out.println("Number");  }  public static void test(Object b1)  {  System.out.println("Object");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  byte |  |  | | --- | | B.  Byte |  |  | | --- | | C.  Number |  |  | | --- | | D.  Object | | | |
|  | | |
| **34.** | class M35  {  public static void test(double b1)  {  System.out.println("double");  }  public static void test(Byte b1)  {  System.out.println("Byte");  }  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(Number b1)  {  System.out.println("Number");  }  public static void test(Object b1)  {  System.out.println("Object");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  Byte |  |  | | --- | | B.  double |  |  | | --- | | C.  Object |  |  | | --- | | D.  Number | | | |
|  | | |

|  |  |
| --- | --- |
| **35.** | class M36  {  public static void test(Byte b1)  {  System.out.println("Byte");  }  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(Number b1)  {  System.out.println("Number");  }  public static void test(Object b1)  {  System.out.println("Object");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  Byte |  |  | | --- | | B.  Integer |  |  | | --- | | C.  Object |  |  | | --- | | D.  Number | | | |
|  | | |
| **36.** | class M37  {  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(Number b1)  {  System.out.println("Number");  }  public static void test(Object b1)  {  System.out.println("Object");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  byte ... |  |  | | --- | | B.  Integer |  |  | | --- | | C.  Object |  |  | | --- | | D.  Number | | | |
|  | | |

|  |  |
| --- | --- |
| **37.** | class M38  {  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(Object b1)  {  System.out.println("Object");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  byte ... |  |  | | --- | | B.  Integer |  |  | | --- | | C.  Object |  |  | | --- | | D.  Compilation Error | | | |
|  | | |
| **38.** | class M39  {  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void test(byte ... b1)  {  System.out.println("byte ...");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  byte ... |  |  | | --- | | B.  Integer |  |  | | --- | | C.  Compilation Error | | | |
|  | | |

|  |  |
| --- | --- |
| **39.** | class M40  {  public static void test(Integer b1)  {  System.out.println("Integer");  }  public static void main(String[] args)  {  byte b1 = 10;  test(b1);  }  } |
| |  | | --- | | A.  Integer |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  Running successful with no output | | | |
|  | | |
| **40.** | For primitive byte data type what is the corresponding wrapper class |
| |  | | --- | | A.  byte |  |  | | --- | | B.  Byte |  |  | | --- | | C.  BYTE |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **41.** | For primitive short data type what is the corresponding wrapper class |
| |  | | --- | | A.  Short |  |  | | --- | | B.  SHORT |  |  | | --- | | C.  short |  |  | | --- | | D.  None | | | |
|  | | |
| **42.** | For primitive int data type what is the corresponding wrapper class |
| |  | | --- | | A.  Int |  |  | | --- | | B.  Integer |  |  | | --- | | C.  integer |  |  | | --- | | D.  int | | | |
|  | | |

|  |  |
| --- | --- |
| **43.** | For primitive long data type what is the corresponding wrapper class |
| |  | | --- | | A.  long |  |  | | --- | | B.  LONG |  |  | | --- | | C.  Long |  |  | | --- | | D.  None | | | |
|  | | |
| **44.** | For primitive float data type what is the corresponding wrapper class |
| |  | | --- | | A.  Float |  |  | | --- | | B.  FLOAT |  |  | | --- | | C.  float |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **45.** | For primitive double data type what is the corresponding wrapper class |
| |  | | --- | | A.  DOUBLE |  |  | | --- | | B.  Double |  |  | | --- | | C.  double |  |  | | --- | | D.  None | | | |
|  | | |
| **46.** | For primitive char data type what is the corresponding wrapper class |
| |  | | --- | | A.  Char |  |  | | --- | | B.  Character |  |  | | --- | | C.  char |  |  | | --- | | D.  character |  |  | | --- | | E.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **47.** | For primitive char boolean type what is the corresponding wrapper class |
| |  | | --- | | A.  boolean |  |  | | --- | | B.  BOOLEAN |  |  | | --- | | C.  Boolean |  |  | | --- | | D.  None | | | |
|  | | |
| **48.** | All wrapper classes are available in……….. |
| |  | | --- | | A.  java.util |  |  | | --- | | B.  java.awt |  |  | | --- | | C.  java.lang |  |  | | --- | | D.  None | | | |
|  | | |

|  |  |
| --- | --- |
| **49.** | Is it possible to develop a sub class to wrapper class |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |
| **50.** | All wrapper classes are not eligible to serilization |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | | |
|  | | |

|  |  |
| --- | --- |
| **51.** | Wrapper classes accepts only objects |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | | |
|  | | |
| **52.** | Is it possible to convert a string into int primitive if string contains characters |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **53.** | What is the special allowed in case of converting string to double primitive or float primitive |
| |  | | --- | | A.  .(dot) |  |  | | --- | | B.  \_(Underscore) |  |  | | --- | | C.  \*(star) |  |  | | --- | | D.  None | | | | |
|  | | | |
| **54.** | While converting string to boolean primitive which is other than true what you will get as output | |
| |  | | --- | | A.  Exception |  |  | | --- | | B.  false |  |  | | --- | | C.  none | | | | |
|  | | | |

|  |  |
| --- | --- |
| **55.** | Inside a character class parse method is not available |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | | |
|  | | |
| **56.** | Auto boxing and auto unboxing introduced in which version |
| |  | | --- | | A.  JDK1.4 |  |  | | --- | | B.  JDK1.3 |  |  | | --- | | C.  JDK1.5 | | | |
|  | | |

|  |  |
| --- | --- |
| **57.** | Every wrapper class can upcast to number or object |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | | |
|  | | |
| **58.** | We can use method arguments as variable number of arguments |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | | |
|  | | |

|  |  |
| --- | --- |
| **59.** | What is the order of execution  i. Widening  ii. Boxing and upcasting  iii. Variable numer of arguments |
| |  | | --- | | A.  i, iii, ii |  |  | | --- | | B.  i, ii, iii |  |  | | --- | | C.  iii, ii, i |  |  | | --- | | D.  ii, i, iii | | | |
|  | | |

Bottom of Form