*Section 8.2 Defining Classes for Objects*

***8.1***  \_\_\_\_\_\_\_\_\_\_ represents an entity in the real world that can be distinctly identified.

 A. A class

 B. An object

 C. A method

 D. A data field



***8.2***  \_\_\_\_\_\_\_ is a construct that defines objects of the same type.

 A. A class

 B. An object

 C. A method

 D. A data field



***8.3***  An object is an instance of a \_\_\_\_\_\_\_\_\_\_.

 A. program

 B. class

 C. method

 D. data



***8.4***  The keyword \_\_\_\_\_\_\_\_\_\_ is required to declare a class.

 A. public

 B. private

 C. class

 D. All of the above.



*Section 8.4 Constructing Objects Using Constructors*

***8.5***  \_\_\_\_\_\_\_\_ is invoked to create an object.

 A. A constructor

 B. The main method

 C. A method with a return type

 D. A method with the void return type



***8.6***  Which of the following statements are true?

 A. A default constructor is provided automatically if no constructors are explicitly declared in the class.

 B. At least one constructor must always be defined explicitly.

 C. Every class has a default constructor.

 D. The default constructor is a no-arg constructor.



***8.7***  Which of the following statements are true?

 A. Multiple constructors can be defined in a class.

 B. Constructors do not have a return type, not even void.

 C. Constructors must have the same name as the class itself.

 D. Constructors are invoked using the new operator when an object is created.



***8.8***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    A a = new A();  
    a.print();  
  }  
}  
  
class A {  
  String s;  
  
  A(String newS) {  
    s = newS;  
  }  
  
  void print() {  
    System.out.println(s);  
  }  
}

 A. The program has a compilation error because class A is not a public class.

 B. The program has a compilation error because class A does not have a no-arg constructor.

 C. The program compiles and runs fine and prints nothing.

 D. The program would compile and run if you change A a = new A() to A a = new A("5").



***8.9***  What is wrong in the following code?  
  
class TempClass {  
  int i;  
  public void TempClass(int j) {  
    int i = j;  
  }  
}  
  
public class C {  
  public static void main(String[] args) {  
    TempClass temp = new TempClass(2);  
  }  
}

 A. The program has a compilation error because TempClass does not have a default constructor.

 B. The program has a compilation error because TempClass does not have a constructor with an int argument.

 C. The program compiles fine, but it does not run because class C is not public.

 D. The program compiles and runs fine.



*Section 8.5 Accessing Objects via Reference Variables*

***8.10***  Given the declaration Circle x = new Circle(), which of the following statement is most accurate.

 A. x contains an int value.

 B. x contains an object of the Circle type.

 C. x contains a reference to a Circle object.

 D. You can assign an int value to x.



***8.11***  Analyze the following code.  
  
public class Test {  
  int x;  
    
  public Test(String t) {  
     System.out.println("Test");  
  }  
  
  public static void main(String[] args) {  
    Test test = null;  
    System.out.println(test.x);  
  }  
}

 A. The program has a compile error because test is not initialized.

 B. The program has a compile error because x has not been initialized.

 C. The program has a compile error because you cannot create an object from the class that defines the object.

 D. The program has a compile error because Test does not have a default constructor.

 E. The program has a runtime NullPointerException because test is null while executing test.x.



***8.12***  The default value for data field of a boolean type, numeric type, object type is \_\_\_\_\_\_\_\_\_\_\_, respectively.

 A. true, 1, Null

 B. false, 0, null

 C. true, 0, null

 D. true, 1, null

 E. false, 1, null



***8.13***  Which of the following statements are true?

 A. Local variables do not have default values.

 B. Data fields have default values.

 C. A variable of a primitive type holds a value of the primitive type.

 D. A variable of a reference type holds a reference to where an object is stored in the memory.

 E. You may assign an int value to a reference variable.



***8.14***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    double radius;  
    final double PI= 3.15169;  
    double area = radius \* radius \* PI;  
    System.out.println("Area is " + area);  
  }  
}

 A. The program has compile errors because the variable radius is not initialized.

 B. The program has a compile error because a constant PI is defined inside a method.

 C. The program has no compile errors but will get a runtime error because radius is not initialized.

 D. The program compiles and runs fine.



***8.15***  Analyze the following code.  
  
public class Test {  
  int x;  
  
  public Test(String t) {  
     System.out.println("Test");  
  }  
  
  public static void main(String[] args) {  
    Test test = new Test();  
    System.out.println(test.x);  
  }  
}

 A. The program has a compile error because System.out.println method cannot be invoked from the constructor.

 B. The program has a compile error because x has not been initialized.

 C. The program has a compile error because you cannot create an object from the class that defines the object.

 D. The program has a compile error because Test does not have a default constructor.



***8.16***  Suppose TestCircle1 and Circle1 in Listing 8.1 are in two separate files named TestCircle1.java and Circle1.java, respectively. What is the outcome of compiling TestCircle.java and then Circle.java?

 A. Only TestCircle1.java compiles.

 B. Only Circle1.java compiles.

 C. Both compile fine.

 D. Neither compiles successfully.



***8.17***  Which of the following statement is most accurate?

 A. A reference variable is an object.

 B. A reference variable refers to an object.

 C. An object may contain other objects.

 D. An object may contain the references of other objects.



*Section 8.6 Using Classes From the Java Library*

***8.18***  The java.util.Date class is introduced in this section. Analyze the following code and choose the best answer:  
  
Which of the following code in A or B, or both creates an object of the Date class:  
  
A:  
public class Test {  
  public Test() {  
    new java.util.Date();  
  }  
}  
  
B:  
public class Test {  
  public Test() {  
    java.util.Date date = new java.util.Date();  
  }  
}

 A. A.

 B. B.

 C. Neither



***8.19***  Which of the following statements are correct?

 A. When creating a Random object, you have to specify the seed or use the default seed.

 B. If two Random objects have the same seed, the sequence of the random numbers obtained from these two objects are identical.

 C. The nextInt() method in the Random class returns the next random int value.

 D. The nextDouble() method in the Random class returns the next random double value.



***8.20***  How many JFrame objects can you create and how many can you display?

 A. one

 B. two

 C. three

 D. unlimited



*Section 8.7 Static Variables, Constants, and Methods*

***8.21***  Variables that are shared by every instances of a class are \_\_\_\_\_\_\_\_\_\_.

 A. public variables

 B. private variables

 C. instance variables

 D. class variables



***8.22***  You should add the static keyword in the place of ? in Line \_\_\_\_\_\_\_\_ in the following code:  
  
1 public class Test {   
2   private int age;  
3   
4   public ? int square(int n) {   
5     return n \* n;  
6   }  
7  
8   public ? int getAge() {  
9  }  
10}

 A. in line 4

 B. in line 8

 C. in both line 4 and line 8

 D. none



***8.23***  A method that is associated with an individual object is called \_\_\_\_\_\_\_\_\_\_.

 A. a static method

 B. a class method

 C. an instance method

 D. an object method



***8.24***  To declare a constant MAX\_LENGTH as a member of the class, you write

 A. final static MAX\_LENGTH = 99.98;

 B. final static float MAX\_LENGTH = 99.98;

 C. static double MAX\_LENGTH = 99.98;

 D. final double MAX\_LENGTH = 99.98;

 E. final static double MAX\_LENGTH = 99.98;



***8.25***  Analyze the following code.  
  
public class Test {  
  public static void main(String[] args) {  
    int n = 2;  
    xMethod(n);  
             
    System.out.println("n is " + n);  
  }  
  
  void xMethod(int n) {  
    n++;  
  }  
}

 A. The code has a compile error because xMethod does not return a value.

 B. The code has a compile error because xMethod is not declared static.

 C. The code prints n is 1.

 D. The code prints n is 2.

 E. The code prints n is 3.



***8.26***  What will be displayed by the second println statement in the main method?  
public class Foo {  
  int i;  
  static int s;  
  
  public static void main(String[] args) {  
    Foo f1 = new Foo();  
    System.out.println("f1.i is " + f1.i + " f1.s is " + f1.s);  
    Foo f2 = new Foo();  
    System.out.println("f2.i is " + f2.i + " f2.s is " + f2.s);  
    Foo f3 = new Foo();  
    System.out.println("f3.i is " + f3.i + " f3.s is " + f3.s);  
  }  
  
  public Foo() {  
    i++;  
    s++;  
  }  
}

 A. f2.i is 1 f2.s is 1

 B. f2.i is 1 f2.s is 2

 C. f2.i is 2 f2.s is 2

 D. f2.i is 2 f2.s is 1



***8.27***  What will be displayed by the third println statement in the main method?   
public class Foo {  
  int i;  
  static int s;  
  
  public static void main(String[] args) {  
    Foo f1 = new Foo();  
    System.out.println("f1.i is " + f1.i + " f1.s is " + f1.s);  
    Foo f2 = new Foo();  
    System.out.println("f2.i is " + f2.i + " f2.s is " + f2.s);  
    Foo f3 = new Foo();  
    System.out.println("f3.i is " + f3.i + " f3.s is " + f3.s);  
  }  
  
  public Foo() {  
    i++;  
    s++;  
  }  
}

 A. f3.i is 1 f3.s is 1

 B. f3.i is 1 f3.s is 2

 C. f3.i is 1 f3.s is 3

 D. f3.i is 3 f3.s is 1

 E. f3.i is 3 f3.s is 3



***8.28***  What code may be filled in the blank without causing syntax or runtime errors:  
  
public class Test {  
  java.util.Date date;  
  
  public static void main(String[] args) {  
    Test test = new Test();  
    System.out.println(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);  
  }  
}

 A. test.date

 B. date

 C. test.date.toString()

 D. date.toString()



***8.29***  Suppose the xMethod() is invoked in the following constructor in a class, xMethod() is \_\_\_\_\_\_\_\_\_ in the class.  
  
public MyClass() {  
  xMethod();  
}

 A. a static method

 B. an instance method

 C. a static method or an instance method



***8.30***  Suppose the xMethod() is invoked from a main method in a class as follows, xMethod() is \_\_\_\_\_\_\_\_\_ in the class.  
  
public static void main(String[] args) {  
  xMethod();  
}

 A. a static method

 B. an instance method

 C. a static method or an instance method



*Section 8.8 Visibility Modifiers*

***8.31***  To prevent a class from being instantiated, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 A. don't use any modifiers on the constructor.

 B. use the public modifier on the constructor.

 C. use the private modifier on the constructor.

 D. use the static modifier on the constructor.



***8.32***  Analyze the following code:  
  
public class Test {  
  public static void main(String args[]) {  
    NClass nc = new NClass();  
    nc.t = nc.t++;  
  }  
}  
  
class NClass {  
  int t;  
  private NClass() {  
  }  
}

 A. The program has a compilation error because the NClass class has a private constructor.

 B. The program does not compile because the parameter list of the main method is wrong.

 C. The program compiles, but has a runtime error because t has no initial value.

 D. The program compiles and runs fine.



***8.33***  Analyze the following code:  
  
public class Test {  
  private int t;  
  
  public static void main(String[] args) {  
    int x;  
    System.out.println(t);  
  }  
}

 A. The variable t is not initialized and therefore causes errors.

 B. The variable t is private and therefore cannot be accessed in the main method.

 C. t is non-static and it cannot be referenced in a static context in the main method.

 D. The variable x is not initialized and therefore causes errors.

 E. The program compiles and runs fine.



***8.34***  Analyze the following code and choose the best answer:  
  
public class Foo {  
  private int x;  
  
  public static void main(String[] args) {  
    Foo foo = new Foo();  
    System.out.println(foo.x);  
  }  
}

 A. Since x is private, it cannot be accessed from an object foo.

 B. Since x is defined in the class Foo, it can be accessed by any method inside the class without using an object. You can write the code to access x without creating an object such as foo in this code.

 C. Since x is an instance variable, it cannot be directly used inside a main method. However, it can be accessed through an object such as foo in this code.

 D. You cannot create a self-referenced object; that is, foo is created inside the class Foo.



*Section 8.9 Data Field Encapsulation*

***8.35***  Which of the following statements are true?

 A. Use the private modifier to encapsulate data fields.

 B. Encapsulating data fields makes the program easy to maintain.

 C. Encapsulating data fields makes the program short.

 D. Encapsulating data fields helps prevent programming errors.



***8.36***  Suppose you wish to provide an accessor method for a boolean property finished, what signature of the method should be?

 A. public void getFinished()

 B. public boolean getFinished()

 C. public boolean isFinished()

 D. public void isFinished()



***8.37***  Which is the advantage of encapsulation?

 A. Only public methods are needed.

 B. Making the class final causes no consequential changes to other code.

 C. It changes the implementation without changing a class's contract and causes no consequential changes to other code.

 D. It changes a class's contract without changing the implementation and causes no consequential changes to other code.



*Section 8.10 Passing Objects to Methods*

***8.38***  When invoking a method with an object argument, \_\_\_\_\_\_\_\_\_\_\_ is passed.

 A. the contents of the object

 B. a copy of the object

 C. the reference of the object

 D. the object is copied, then the reference of the copied object



***8.39***  What is the value of myCount.count displayed?  
public class Test {  
  public static void main(String[] args) {  
    Count myCount = new Count();  
    int times = 0;  
  
    for (int i=0; i<100; i++)  
      increment(myCount, times);  
  
    System.out.println(  
      "myCount.count = " + myCount.count);  
    System.out.println("times = "+ times);  
  }  
  
  public static void increment(Count c, int times) {  
    c.count++;  
    times++;  
  }  
}  
  
class Count {  
  int count;  
  
  Count(int c) {  
    count = c;  
  }  
  
  Count() {  
    count = 1;  
  }  
}

 A. 101

 B. 100

 C. 99

 D. 98



***8.40***  What is the value of times displayed?  
public class Test {  
  public static void main(String[] args) {  
    Count myCount = new Count();  
    int times = 0;  
  
    for (int i=0; i<100; i++)  
      increment(myCount, times);  
  
    System.out.println(  
      "myCount.count = " + myCount.count);  
    System.out.println("times = "+ times);  
  }  
  
  public static void increment(Count c, int times) {  
    c.count++;  
    times++;  
  }  
}  
  
class Count {  
  int count;  
  
  Count(int c) {  
    count = c;  
  }  
  
  Count() {  
    count = 1;  
  }  
}

 A. 101

 B. 100

 C. 99

 D. 98

 E. 0



***8.41***  What is the output of the following program?  
  
import java.util.Date;  
  
public class Test {  
  public static void main(String[] args) {  
    Date date = new Date(1234567);  
    m1(date);  
    System.out.print(date.getTime() + " ");  
  
    m2(date);  
    System.out.println(date.getTime());  
  }  
  
  public static void m1(Date date) {  
    date = new Date(7654321);  
  }  
  
  public static void m2(Date date) {  
    date.setTime(7654321);  
  }  
}

 A. 1234567 1234567

 B. 1234567 7654321

 C. 7654321 1234567

 D. 7654321 7654321



*Section 8.11 Array of Objects*

***8.42***  Given the declaration Circle[] x = new Circle[10], which of the following statement is most accurate?

 A. x contains an array of ten int values.

 B. x contains an array of ten objects of the Circle type.

 C. x contains a reference to an array and each element in the array can hold a reference to a Circle object.

 D. x contains a reference to an array and each element in the array can hold a Circle object.



***8.43***  Assume java.util.Date[] dates = new java.util.Date[10], which of the following statements are true?

 A. dates is null.

 B. dates[0] is null.

 C. dates = new java.util.Date[5] is fine, which assigns a new array to dates.

 D. dates = new Date() is fine, which creates a new Date object and assigns to dates.



*Section 10.2 Immutable Objects and Classes*

***10.1***  Which of the following statements are true about an immutable object?

 A. The contents of an immutable object cannot be modified.

 B. All properties of an immutable object must be private.

 C. All properties of an immutable object must be of primitive types.

 D. An object type property in an immutable object must also be immutable.

 E. An immutable object contains no mutator methods.

Your answer AE is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABDE

*Section 10.3 Scope of Variables*

***10.2***  What is the printout for the first statement in the main method?  
public class Foo {  
  static int i = 0;  
  static int j = 0;  
  
  public static void main(String[] args) {  
    int i = 2;  
    int k = 3;  
    {  
      int j = 3;  
      System.out.println("i + j is " + i + j);  
    }  
  
    k = i + j;  
    System.out.println("k is " + k);  
    System.out.println("j is " + j);  
  }  
}

 A. i + j is 5

 B. i + j is 6

 C. i + j is 22

 D. i + j is 23

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D  
Explanation: The first + operator in the expression 'i + j is ' + i + j is evaluated.

***10.3***  What will be displayed from the statement System.out.println("k is " + k) in the main method?  
public class Foo {  
  static int i = 0;  
  static int j = 0;  
  
  public static void main(String[] args) {  
    int i = 2;  
    int k = 3;  
    {  
      int j = 3;  
      System.out.println("i + j is " + i + j);  
    }  
  
    k = i + j;  
    System.out.println("k is " + k);  
    System.out.println("j is " + j);  
  }  
}

 A. k is 0

 B. k is 1

 C. k is 2

 D. k is 3

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***10.4***  What will be displayed from the statement System.out.println("j is " + j) in the main method?  
public class Foo {  
  static int i = 0;  
  static int j = 0;  
  
  public static void main(String[] args) {  
    int i = 2;  
    int k = 3;  
    {  
      int j = 3;  
      System.out.println("i + j is " + i + j);  
    }  
  
    k = i + j;  
    System.out.println("k is " + k);  
    System.out.println("j is " + j);  
  }  
}

 A. j is 0

 B. j is 1

 C. j is 2

 D. j is 3

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***10.5***  You can declare two variables with the same name in \_\_\_\_\_\_\_\_\_\_.

 A. a method one as a formal parameter and the other as a local variable

 B. a block

 C. two nested blocks in a method (two nested blocks means one being inside the other)

 D. different methods in a class

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

Click here to show the correct answer

*Section 10.4 The this Keyword*

***10.6***  Analyze the following code:  
  
class Circle {  
  private double radius;  
    
  public Circle(double radius) {  
    radius = radius;  
  }  
}

 A. The program has a compilation error because it does not have a main method.

 B. The program will compile, but you cannot create an object of Circle with a specified radius. The object will always have radius 0.

 C. The program has a compilation error because you cannot assign radius to radius.

 D. The program does not compile because Circle does not have a default constructor.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***10.7***  Analyze the following code:  
  
class Test {  
  private double i;  
    
  public Test(double i) {  
    this.t();  
    this.i = i;  
  }  
  
  public Test() {  
    System.out.println("Default constructor");  
    this(1);  
  }  
  
  public void t() {  
    System.out.println("Invoking t");  
  }  
}

 A. this.t() may be replaced by t().

 B. this.i may be replaced by i.

 C. this(1) must be called before System.out.println("Default constructor").

 D. this(1) must be replaced by this(1.0).

Your answer AB is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is AC

***10.8***  Which of the following can be placed in the blank line in the following code?  
public class Test {  
  private int id;  
    
  public void m1() {  
    \_\_\_\_\_.id = 45;  
  }  
}

 A. this

 B. Test

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***10.9***  Analyze the following code:  
public class Test {  
  public static void main(String[] args) {  
    String firstName = "John";  
    Name name = new Name(firstName, 'F', "Smith");  
    firstName = "Peter";  
    name.lastName = "Pan";  
    System.out.println(name.firstName + " " + name.lastName);  
  }  
}  
  
class Name {  
  String firstName;  
  char mi;  
  String lastName;  
    
  public Name(String firstName, char mi, String lastName) {  
    this.firstName = firstName;  
    this.mi = mi;  
    this.lastName = lastName;  
  }  
}

 A. The program displays Peter Pan.

 B. The program displays John Pan.

 C. The program displays Peter Smith.

 D. The program displays John Smith.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***10.10***  Analyze the following code:  
public class Test {  
  public static void main(String[] args) {  
    MyDate birthDate = new MyDate(1990, 5, 6);  
    Name name = new Name("John", 'F', "Smith", birthDate);  
    birthDate = new MyDate(1991, 1, 1);  
    birthDate.year = 1992;  
    System.out.println(name.birthDate.year);  
  }  
}  
  
class MyDate {  
  int year;  
  int month;  
  int day;  
    
  public MyDate(int year, int month, int day) {  
    this.year = year;  
    this.month = month;  
    this.day = day;  
  }  
}  
  
class Name {  
  String firstName;  
  char mi;  
  String lastName;  
  MyDate birthDate;  
    
  public Name(String firstName, char mi, String lastName, MyDate birthDate) {  
    this.firstName = firstName;  
    this.mi = mi;  
    this.lastName = lastName;  
    this.birthDate = birthDate;  
  }  
}

 A. The program displays 1990.

 B. The program displays 1991.

 C. The program displays 1992.

 D. The program displays no thing.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

*Section 10.7 Object Composition*

***10.11***  \_\_\_\_\_\_\_\_\_\_\_ is attached to the class of the composing class to denote the aggregation relationship with the composed object.

 A. An empty diamond

 B. A solid diamond

 C. An empty oval

 D. A solid oval

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***10.12***  An aggregation relationship is usually represented as \_\_\_\_\_\_\_\_\_\_ in \_\_\_\_\_\_\_\_\_\_\_.

 A. a data field/the aggregating class

 B. a data field/the aggregated class

 C. a method/the aggregating class

 D. a method/the aggregated class

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

*Section 10.11 Class Design Guidelines*

***10.13***  Which of the following statements are true?

 A. A class should describe a single entity and all the class operations should logically fit together to support a coherent purpose.

 B. A class should always contain a no-arg constructor.

 C. The constructors must always be public.

 D. The constructors may be protected.

Your answer AC is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is AD  
Explanation: (B) is not true. Most classes have a no-arg constructor. But sometimes, it does not make any sense to provide a no-arg constructor. For example, StringTokenizer does not have a no-arg constructor. (C) is not true. For example, the constructor in the Math class is private to prevent creating instances from the class, because there is no need to create instances for the Math class since all methods are static in Math. The constructors for abstract classes should be protected in most cases.

***10.14***  Which of the following is poor design?

 A. A data field is derived from other data fields in the same class.

 B. A method must be invoked after/before invoking another method in the same class.

 C. A method is an instance method, but it does not reference any instance data fields or invoke instance methods.

 D. A parameter is passed from a constructor to initialize a static data field.

Your answer AC is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCD

*Section 10.12 Processing Primitive Data Type Values as Objects*

***10.15***  Which of the following statements will convert a string s into i of int type?

 A. i = Integer.parseInt(s);

 B. i = (new Integer(s)).intValue();

 C. i = Integer.valueOf(s).intValue();

 D. i = Integer.valueOf(s);

 E. i = (int)(Double.parseDouble(s));

Your answer BD is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCE

***10.16***  Which of the following statements will convert a string s into a double value d?

 A. d = Double.parseDouble(s);

 B. d = (new Double(s)).doubleValue();

 C. d = Double.valueOf(s).doubleValue();

 D. All of the above.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D  
Explanation: All are fine. a is preferred because it does not have to create an object.

***10.17***  Which of the following statements convert a double value d into a string s?

 A. s = (new Double(d)).toString();

 B. s = (Double.valueOf(s)).toString();

 C. s = new Double(d).stringOf();

 D. s = String.stringOf(d);

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***10.18***  Which of the following statements is correct?

 A. Integer.parseInt("12", 2);

 B. Integer.parseInt(100);

 C. Integer.parseInt("100");

 D. Integer.parseInt(100, 16);

 E. Integer.parseInt("345", 8);

Your answer AB is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is CE  
Explanation: (A) is incorrect because 12 is not a binary number. (B) and (D) are incorrect because the first argument in the parseInt method must be a string.

***10.19***  What is the output of Integer.parseInt("10", 2)?

 A. 1;

 B. 2;

 C. 10;

 D. Invalid statement;

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B  
Explanation: Based on 2, 10 is 2 in decimal.

*Section 10.13 Automatic Conversion Between Primitive Types and Wrapper Class Types*

***10.20***  In JDK 1.5, you may directly assign a primitive data type value to a wrapper object. This is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

 A. auto boxing

 B. auto unboxing

 C. auto conversion

 D. auto casting

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

Click here to show the correct answer

***10.21***  In JDK 1.5, analyze the following code.  
  
Line 1: Integer[] intArray = {1, 2, 3};  
Line 2: int i = intArray[0] + intArray[1];  
Line 3: int j = i + intArray[2];  
Line 4: double d = intArray[0];

 A. It is OK to assign 1, 2, 3 to an array of Integer objects in JDK 1.5.

 B. It is OK to automatically convert an Integer object to an int value in Line 2.

 C. It is OK to mix an int value with an Integer object in an expression in Line 3.

 D. Line 4 is OK. An int value from intArray[0] object is assigned to a double variable d.

Your answer BC is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCD

*Section 10.14 The BigInteger and BigDecimal Classes*

***10.22***  To create an instance of BigInteger for 454, use

 A. BigInteger(454);

 B. new BigInteger(454);

 C. BigInteger("454");

 D. new BigInteger("454");

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***10.23***  To create an instance of BigDecimal for 454.45, use

 A. BigInteger(454.45);

 B. new BigInteger(454.45);

 C. BigInteger("454.45");

 D. new BigDecimal("454.45");

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***10.24***  BigInteger and BigDecimal are immutable

 A. true

 B. false

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***10.25***  To add BigInteger b1 to b2, you write \_\_\_\_\_\_\_\_\_.

 A. b1.add(b2);

 B. b2.add(b1);

 C. b2 = b1.add(b2);

 D. b2 = b2.add(b1);

 E. b1 = b2.add(b1);

Your answer ABC is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is CD

***10.26***  What is the output of the following code?  
  
public class Test {  
  public static void main(String[] args) {  
    java.math.BigInteger x = new java.math.BigInteger("3");  
    java.math.BigInteger y = new java.math.BigInteger("7");  
    x.add(y);  
    System.out.println(x);  
  }  
}

 A. 3

 B. 4

 C. 10

 D. 11

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***10.27***  To divide BigDecimal b1 by b2 and assign the result to b1, you write \_\_\_\_\_\_\_\_\_.

 A. b1.divide(b2);

 B. b2.divide(b1);

 C. b1 = b1.divide(b2);

 D. b1 = b2.divide(b1);

 E. b1 = b2.divide(b1);

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***10.28***  Which of the following classes are immutable?

 A. Integer

 B. Double

 C. BigInteger

 D. BigDecimal

 E. String

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCDE

***10.29***  Which of the following statements are correct?

 A. new java.math.BigInteger("343");

 B. new java.math.BigDecimal("343.445");

 C. new java.math.BigInteger(343);

 D. new java.math.BigDecimal(343.445);

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is AB

*Section 11.2 Superclasses and Subclasses*

***11.1***  Object-oriented programming allows you to derive new classes from existing classes. This is called \_\_\_\_\_\_\_\_\_\_\_\_.

 A. encapsulation

 B. inheritance

 C. abstraction

 D. generalization

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.2***  Which of the following statements are true?

 A. A subclass is a subset of a superclass.

 B. A subclass is usually extended to contain more functions and more detailed information than its superclass.

 C. "class A extends B" means A is a subclass of B.

 D. "class A extends B" means B is a subclass of A.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is BC

***11.3***  What is the output of the following code?  
public class Test1 {  
  public static void main(String[] args) {  
    ChildClass c = new ChildClass();  
    c.print();  
  }  
}  
  
class ParentClass {  
  int id = 1;  
  void print() {  
    System.out.println(id);  
  }  
}  
  
class ChildClass extends ParentClass {  
  int id = 2;  
}

 A. 0

 B. 1

 C. 2

 D. Nothing

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

*Section 11.3 Using the super KeywordSection 11.3.1 Calling Superclass Constructors*

***11.4***  Suppose you create a class Square to be a subclass of GeometricObject. Analyze the following code:  
  
class Square extends GeometricObject {  
  double length;  
    
  Square(double length) {  
    GeometricObject(length);  
  }  
}

 A. The program compiles fine, but you cannot create an instance of Square because the constructor does not specify the length of the Square.

 B. The program has a compile error because you attempted to invoke the GeometricObject class's constructor illegally.

 C. The program compiles fine, but it has a runtime error because of invoking the Square class's constructor illegally.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.5***  Analyze the following code:  
  
  
public class A extends B {  
}  
  
class B {  
  public B(String s) {  
  }  
}

 A. The program has a compilation error because A does not have a default constructor.

 B. The program has a compilation error because the default constructor of A invokes the default constructor of B, but B does not have a default constructor.

 C. The program would compile fine if you add the following constructor into A: A(String s) { }

 D. The program would compile fine if you add the following constructor into A: A(String s) { super(s); }

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is BD

***11.6***  Analyze the following code:  
  
public class Test extends A {   
  public static void main(String[] args) {  
    Test t = new Test();  
    t.print();  
  }  
}  
  
class A {  
  String s;  
  
  A(String s) {  
    this.s = s;  
  }  
  
  public void print() {  
    System.out.println(s);  
  }  
}

 A. The program does not compile because Test does not have a default constructor Test().

 B. The program has an implicit default constructor Test(), but it cannot be compiled, because its super class does not have a default constructor. The program would compile if the constructor in the class A were removed.

 C. The program would compile if a default constructor A(){ } is added to class A explicitly.

 D. The program compiles, but it has a runtime error due to the conflict on the method name print.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is BC  
Explanation: See the last Note in the section, 'Using the super keyword.'

*Section 11.3.2 Constructor Chaining*

***11.7***  What is the output of running class C?  
  
class A {  
  public A() {  
    System.out.println(  
      "The default constructor of A is invoked");  
  }  
}  
  
class B extends A {  
  public B() {  
    System.out.println(  
      "The default constructor of B is invoked");  
  }  
}  
  
public class C  {  
  public static void main(String[] args) {  
    B b = new B();  
  }  
}

 A. Nothing displayed

 B. "The default constructor of B is invoked"

 C. "The default constructor of A is invoked""The default constructor of B is invoked"

 D. "The default constructor of B is invoked""The default constructor of A is invoked"

 E. "The default constructor of A is invoked"

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.8***  Which of the following is incorrect?

 A. A constructor may be static.

 B. A constructor may be private.

 C. A constructor may invoke a static method.

 D. A constructor may invoke an overloaded constructor.

 E. A constructor invokes its superclass no-arg constructor by default if a constructor does not invoke an overloaded constructor or its superclass?s constructor.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

*Section 11.3.3 Calling Superclass Methods*

***11.9***  Which of the statements regarding the super keyword is incorrect?

 A. You can use super to invoke a super class constructor.

 B. You can use super to invoke a super class method.

 C. You can use super.super.p to invoke a method in superclass's parent class.

 D. You cannot invoke a method in superclass's parent class.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

*Section 11.4 Overriding Methods*

***11.10***  Analyze the following code:  
  
public class Test {   
  public static void main(String[] args) {  
    B b = new B();  
    b.m(5);  
    System.out.println("i is " + b.i);  
  }  
}  
  
class A {  
  int i;  
  
  public void m(int i) {  
    this.i = i;  
  }  
}  
  
class B extends A {  
  public void m(String s) {  
  }  
}

 A. The program has a compilation error, because m is overridden with a different signature in B.

 B. The program has a compilation error, because b.m(5) cannot be invoked since the method m(int) is hidden in B.

 C. The program has a runtime error on b.i, because i is not accessible from b.

 D. The method m is not overridden in B. B inherits the method m from A and defines an overloaded method m in B.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***11.11***  The getValue() method is overridden in two ways. Which one is correct?  
  
I:  
public class Test {  
  public static void main(String[] args) {  
    A a = new A();  
    System.out.println(a.getValue());  
  }  
}  
  
class B {  
  public String getValue() {  
    return "Any object";  
  }  
}  
  
class A extends B {  
  public Object getValue() {  
    return "A string";  
  }  
}  
  
II:  
public class Test {  
  public static void main(String[] args) {  
    A a = new A();  
    System.out.println(a.getValue());  
  }  
}  
  
class B {  
  public Object getValue() {  
    return "Any object";  
  }  
}  
  
class A extends B {  
  public String getValue() {  
    return "A string";  
  }  
}

 A. I

 B. II

 C. Both I and II

 D. Neither

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

*Section 11.5 Overriding vs. Overloading*

***11.12***  Which of the following statements are true?

 A. To override a method, the method must be defined in the subclass using the same signature and compatible return type as in its superclass.

 B. Overloading a method is to provide more than one method with the same name but with different signatures to distinguish them.

 C. It is a compilation error if two methods differ only in return type in the same class.

 D. A private method cannot be overridden. If a method defined in a subclass is private in its superclass, the two methods are completely unrelated.

 E. A static method cannot be overridden. If a static method defined in the superclass is redefined in a subclass, the method defined in the superclass is hidden.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCDE

***11.13***  Which of the following statements are true?

 A. A method can be overloaded in the same class.

 B. A method can be overridden in the same class.

 C. If a method overloads another method, these two methods must have the same signature.

 D. If a method overrides another method, these two methods must have the same signature.

 E. A method in a subclass can overload a method in the superclass.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ADE

***11.14***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    new B();  
  }  
}  
  
class A {  
  int i = 7;  
    
  public A() {  
    System.out.println("i from A is " + i);  
  }  
    
  public void setI(int i) {  
    this.i = 2 \* i;  
  }  
}  
  
class B extends A {  
  public B() {  
    setI(20);  
    // System.out.println("i from B is " + i);  
  }  
    
  public void setI(int i) {  
    this.i = 3 \* i;  
  }  
}

 A. The constructor of class A is not called.

 B. The constructor of class A is called and it displays "i from A is 7".

 C. The constructor of class A is called and it displays "i from A is 40".

 D. The constructor of class A is called and it displays "i from A is 60".

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.15***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    new B();  
  }  
}  
  
class A {  
  int i = 7;  
    
  public A() {  
    setI(20);  
    System.out.println("i from A is " + i);  
  }  
    
  public void setI(int i) {  
    this.i = 2 \* i;  
  }  
}  
  
class B extends A {  
  public B() {  
    // System.out.println("i from B is " + i);  
  }  
    
  public void setI(int i) {  
    this.i = 3 \* i;  
  }  
}

 A. The constructor of class A is not called.

 B. The constructor of class A is called and it displays "i from A is 7".

 C. The constructor of class A is called and it displays "i from A is 40".

 D. The constructor of class A is called and it displays "i from A is 60".

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

*Section 11.6 The Object Class and Its toString() Method*

***11.16***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    Object a1 = new A();  
    Object a2 = new Object();  
    System.out.println(a1);  
    System.out.println(a2);  
  }  
}  
  
class A {  
  int x;  
  
  public String toString() {  
    return "A's x is " + x;  
  }  
}

 A. The program cannot be compiled, because System.out.println(a1) is wrong and it should be replaced by System.out.println(a1.toString());

 B. When executing System.out.println(a1), the toString() method in the Object class is invoked.

 C. When executing System.out.println(a2), the toString() method in the Object class is invoked.

 D. When executing System.out.println(a1), the toString() method in the A class is invoked.

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is CD  
Explanation: Since a1 is an instance of A, the toString() method in the A class is invoked at runtime.

*Sections 11.7-11.8*

***11.17***  Which of the following statements are true?

 A. You can always pass an instance of a subclass to a parameter of its superclass type. This feature is known as polymorphism.

 B. The compiler finds a matching method according to parameter type, number of parameters, and order of the parameters at compilation time.

 C. A method may be implemented in several subclasses. The Java Virtual Machine dynamically binds the implementation of the method at runtime.

 D. Dynamic binding can apply to static methods.

 E. Dynamic binding can apply to instance methods.

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCE

***11.18***  Given the following code, find the compile error?  
  
public class Test {  
  public static void main(String[] args) {  
    m(new GraduateStudent());  
    m(new Student());  
    m(new Person());  
    m(new Object());  
  }  
  
  public static void m(Student x) {  
    System.out.println(x.toString());  
  }  
}  
  
class GraduateStudent extends Student {  
}  
  
class Student extends Person {  
  public String toString() {  
    return "Student";  
  }  
}  
  
class Person extends Object {  
  public String toString() {  
    return "Person";  
  }  
}

 A. m(new GraduateStudent()) causes an error

 B. m(new Student()) causes an error

 C. m(new Person()) causes an error

 D. m(new Object()) causes an error

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is CD

***11.19***  What is the output of the following code?  
  
public class Test {  
  public static void main(String[] args) {  
    new Person().printPerson();  
    new Student().printPerson();  
  }  
}  
  
class Student extends Person {  
  public String getInfo() {  
    return "Student";  
  }  
}  
  
class Person {  
  public String getInfo() {  
    return "Person";  
  }  
    
  public void printPerson() {  
    System.out.println(getInfo());  
  }  
}

 A. Person Person

 B. Person Student

 C. Student Student

 D. Student Person

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.20***  What is the output of the following code?  
  
public class Test {  
  public static void main(String[] args) {  
    new Person().printPerson();  
    new Student().printPerson();  
  }  
}  
  
class Student extends Person {  
  private String getInfo() {  
    return "Student";  
  }  
}  
  
class Person {  
  private String getInfo() {  
    return "Person";  
  }  
    
  public void printPerson() {  
    System.out.println(getInfo());  
  }  
}

 A. Person Person

 B. Person Student

 C. Student Student

 D. Student Person

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

*Section 11.9 Casting Objects and the instanceof Operator*

***11.21***  Which of the following are Java keywords?

 A. instanceOf

 B. instanceof

 C. cast

 D. casting

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.22***  Analyze the following code:  
  
Cylinder cy = new Cylinder(1, 1);  
Circle c = cy;

 A. The code has a compile error.

 B. The code has a runtime error.

 C. The code is fine.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C

***11.23***  Analyze the following code:  
  
Circle c = new Circle (5);  
Cylinder c = cy;

 A. The code has a compile error.

 B. The code has a runtime error.

 C. The code is fine.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***11.24***  Given the following classes and their objects:  
  
class C1 {};  
class C2 extends C1 {};  
class C3 extends C1 {};  
  
C2 c2 = new C2();  
C3 c3 = new C3();  
  
Analyze the following statement:  
  
c2 = (C2)((C1)c3);

 A. c3 is cast into c2 successfully.

 B. You will get a runtime error because you cannot cast objects from sibling classes.

 C. You will get a runtime error because the Java runtime system cannot perform multiple casting in nested form.

 D. The statement is correct.

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.25***  Given the following code:  
  
class C1 {}  
class C2 extends C1 { }  
class C3 extends C2 { }  
class C4 extends C1 {}  
  
C1 c1 = new C1();  
C2 c2 = new C2();  
C3 c3 = new C3();  
C4 c4 = new C4();  
  
Which of the following expressions evaluates to false?

 A. c1 instanceof C1

 B. c2 instanceof C1

 C. c3 instanceof C1

 D. c4 instanceof C2

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***11.26***  Analyze the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    String s = new String("Welcome to Java");  
    Object o = s;  
    String d = (String)o;  
  }  
}

 A. When assigning s to o in Object o = s, a new object is created.

 B. When casting o to s in String d = (String)o, a new object is created.

 C. When casting o to s in String d = (String)o, the contents of o is changed.

 D. s, o, and d reference the same String object.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D  
Explanation: Casting object reference variable does not affect the contents of the object.

***11.27***  You can assign \_\_\_\_\_\_\_\_\_ to a variable of Object[] type.

 A. new char[100]

 B. new int[100]

 C. new double[100]

 D. new String[100]

 E. new java.util.Date[100]

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is DE  
Explanation: Primitive data type array is not compatible with Object[].

*Section 11.10 The Object?s equals() Method*

***11.28***  The equals method is defined in the Object class. Which of the following is correct to override it in the String class?

 A. public boolean equals(String other)

 B. public boolean equals(Object other)

 C. public static boolean equals(String other)

 D. public static boolean equals(Object other)

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.29***  Which of the following statements are true?

 A. Override the toString method defined in the Object class whenever possible.

 B. Override the equals method defined in the Object class whenever possible.

 C. A public default no-arg constructor is assumed if no constructors are defined explicitly.

 D. You should follow standard Java programming style and naming conventions. Choose informative names for classes, data fields, and methods.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCD

***11.30***  What is the output of the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    Object o1 = new Object();  
    Object o2 = new Object();  
    System.out.print((o1 == o2) + " " + (o1.equals(o2)));  
  }  
}

 A. false false

 B. true true

 C. false true

 D. true false

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A  
Explanation: o1 == o2 is false, since o1 and o2 are two different objects. o1.equals(o2) is false since the equals method returns o1 == o2 in the Object class.

***11.31***  What is the output of the following code:  
  
public class Test {  
  public static void main(String[] args) {  
    String s1 = new String("Java");  
    String s2 = new String("Java");  
    System.out.print((s1 == s2) + " " + (s1.equals(s2)));  
  }  
}

 A. false false

 B. true true

 C. false true

 D. true false

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C  
Explanation: s1 == s2 is false, since s1 and s2 are two different objects. s1.equals(s2) is true since the equals method returns true if two strings have the same content.

***11.32***  Given two reference variables t1 and t2, if t1 == t2 is true, t1.equals(t2) must be \_\_\_\_\_\_\_\_\_\_\_.

 A. false

 B. true

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.33***  Given two reference variables t1 and t2, if t1.equals(t2) is true, t1 == t2 \_\_\_\_\_\_\_\_\_\_\_.

 A. is always false

 B. is always true

 C. may be true or false

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.34***  Analyze the following code.  
  
// Program 1:  
public class Test {  
  public static void main(String[] args) {  
    Object a1 = new A();  
    Object a2 = new A();  
    System.out.println(a1.equals(a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(Object a) {  
    return this.x == ((A)a).x;  
  }  
}  
  
  
// Program 2:  
public class Test {  
  public static void main(String[] args) {  
    Object a1 = new A();  
    Object a2 = new A();  
    System.out.println(a1.equals(a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(A a) {  
    return this.x == a.x;  
  }  
}

 A. Program 1 displays true and Program 2 displays true

 B. Program 1 displays false and Program 2 displays true

 C. Program 1 displays true and Program 2 displays false

 D. Program 1 displays false and Program 2 displays false

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C  
Explanation: In Program 1, the equals method in the Object class is overridden. a1.equals(a2) invokes this method. It returns true. In Program 2, the equals method in the Object class is not overridden. a1.equals(a2) invokes the equals method defined in the Object class, which returns false in this case.

***11.35***  Analyze the following code.  
  
// Program 1:  
public class Test {  
  public static void main(String[] args) {  
    Object a1 = new A();  
    Object a2 = new A();  
    System.out.println(a1.equals(a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(A a) {  
    return this.x == a.x;  
  }  
}  
  
  
// Program 2:  
public class Test {  
  public static void main(String[] args) {  
    A a1 = new A();  
    A a2 = new A();  
    System.out.println(a1.equals(a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(A a) {  
    return this.x == a.x;  
  }  
}

 A. Program 1 displays true and Program 2 displays true

 B. Program 1 displays false and Program 2 displays true

 C. Program 1 displays true and Program 2 displays false

 D. Program 1 displays false and Program 2 displays false

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B  
Explanation: In Program 1, the equals method in the Object class is invoked. In Program 2, the equals method in the class A is invoked. There are now two overloaded methods available in the class A. i.e. public boolean equals(Object a) and public boolean equals(A a). Which of the two is used by a1.equals(a2) is determined at compilation time. a1.equals(a2) in Program 1 matches the equals method defined in Object and a1.equals(a2) in Program 2 matches the equals method defined in the class A.

***11.36***  Analyze the following code.  
  
// Program 1  
public class Test {  
  public static void main(String[] args) {  
    Object a1 = new A();  
    Object a2 = new A();  
    System.out.println(((A)a1).equals((A)a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(A a) {  
    return this.x == a.x;  
  }  
}  
  
  
// Program 2  
public class Test {  
  public static void main(String[] args) {  
    A a1 = new A();  
    A a2 = new A();  
    System.out.println(a1.equals(a2));  
  }  
}  
  
class A {  
  int x;  
  
  public boolean equals(A a) {  
    return this.x == a.x;  
  }  
}

 A. Program 1 displays true and Program 2 displays true

 B. Program 1 displays false and Program 2 displays true

 C. Program 1 displays true and Program 2 displays false

 D. Program 1 displays false and Program 2 displays false

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A  
Explanation: In Program 1, ((A)a1).equals((A)a2) matches the equals(A a) method in the class A.

*Section 11.11 The ArrayList Class*

***11.37***  You can create an ArrayList using \_\_\_\_\_\_\_\_\_.

 A. new ArrayList[]

 B. new ArrayList[100]

 C. new ArrayList()

 D. ArrayList()

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.38***  Invoking \_\_\_\_\_\_\_\_\_ removes all elements in an ArrayList x.

 A. x.remove()

 B. x.clean()

 C. x.delete()

 D. x.empty()

 E. x.clear()

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is E

***11.39***  Suppose ArrayList x contains two strings [Beijing, Singapore]. Which of the following methods will cause the list to become [Beijing, Chicago, Singapore]?

 A. x.add("Chicago")

 B. x.add(0, "Chicago")

 C. x.add(1, "Chicago")

 D. x.add(2, "Chicago")

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.40***  Suppose ArrayList x contains two strings [Beijing, Singapore]. Which of the following method will cause the list to become [Beijing]?

 A. x.remove("Singapore")

 B. x.remove(0)

 C. x.remove(1)

 D. x.remove(2)

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is AC

***11.41***  Suppose ArrayList x contains two strings [Beijing, Singapore]. Which of the following method will cause runtime errors?

 A. x.get(1)

 B. x.set(2, "New York");

 C. x.get(2)

 D. x.remove(2)

 E. x.size()

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is BCD  
Explanation: There is no element at index 2.

***11.42***  Invoking \_\_\_\_\_\_\_\_\_ returns the first element in an ArrayList x.

 A. x.first()

 B. x.get(0)

 C. x.get(1)

 D. x.get()

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.43***  Invoking \_\_\_\_\_\_\_\_\_ returns the number of the elements in an ArrayList x.

 A. x.getSize()

 B. x.getLength(0)

 C. x.length(1)

 D. x.size()

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***11.44***  Analyze the following code:  
  
ArrayList<String> list = new ArrayList<String>();  
list.add("Beijing");  
list.add("Tokyo");  
list.add("Shanghai");  
list.set(3, "Hong Kong");

 A. The last line in the code causes a runtime error because there is no element at index 3 in the array list.

 B. The last line in the code has a compile error because there is no element at index 3 in the array list.

 C. If you replace the last line by list.add(3, "Hong Kong"), the code will compile and run fine.

 D. If you replace the last line by list.add(4, "Hong Kong"), the code will compile and run fine.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is AC  
Explanation: There is no element at index 3.

***11.45***  What is output of the following code:  
  
    ArrayList<java.util.Date> list = new ArrayList<java.util.Date>();  
    java.util.Date d = new java.util.Date();  
    list.add(d);  
    list.add(d);  
    System.out.println((list.get(0) == list.get(1)) + " " + (list.get(0)).equals(list.get(1)));

 A. true false

 B. false true

 C. true true

 D. false false

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.46***  What is output of the following code:  
  
    ArrayList<String> list = new ArrayList<String>();  
    String s1 = new String("Java");  
    String s2 = new String("Java");  
    list.add(s1);  
    list.add(s2);  
    System.out.println((list.get(0) == list.get(1)) + " " + (list.get(0)).equals(list.get(1)));

 A. true false

 B. false true

 C. true true

 D. false false

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B  
Explanation: list.get(0) and list.get(1) point to two different objects with the same string contents.

***11.47***  Suppose an ArrayList list contains {"red", "green", "red", "green"}. What is list after the following code?  
  
    list.remove("red");

 A. {"red", "green", "red", "green"}

 B. {"green", "red", "green"}

 C. {"green", "green"}

 D. {"red", "green", "green"}

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.48***  Suppose an ArrayList list contains {"red", "red", "green"}. What is list after the following code?  
  
    String element = "red";  
    for (int i = 0; i < list.size(); i++)  
      if (list.get(i).equals(element))  
        list.remove(element);

 A. {"red", "red", "green"}

 B. {"red", "green"}

 C. {"green"}

 D. {}

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.49***  Suppose an ArrayList list contains {"red", "red", "green"}. What is list after the following code?  
  
    String element = "red";  
    for (int i = 0; i < list.size(); i++)  
      if (list.get(i).equals(element)) {  
        list.remove(element);  
        i--;  
      }

 A. {"red", "red", "green"}

 B. {"red", "green"}

 C. {"green"}

 D. {}

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C

***11.50***  Suppose an ArrayList list contains {"red", "red", "green"}. What is list after the following code?  
  
    String element = "red";  
    for (int i = list.size() - 1; i >= 0; i--)  
      if (list.get(i).equals(element))  
        list.remove(element);

 A. {"red", "red", "green"}

 B. {"red", "green"}

 C. {"green"}

 D. {}

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.51***  The output from the following code is \_\_\_\_\_\_\_\_\_\_.  
  
java.util.ArrayList<String> list = new java.util.ArrayList<String>();  
list.add("New York");   
java.util.ArrayList<String> list1 = list;  
list.add("Atlanta");   
list1.add("Dallas");   
System.out.println(list1);

 A. [New York]

 B. [New York, Atlanta]

 C. [New York, Atlanta, Dallas]

 D. [New York, Dallas]

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C

***11.52***  Which of the following code is corrcet?

 A. ArrayList<Integer> list = new ArrayList<>(); list.add(3.4);

 B. ArrayList<Double> list = new ArrayList<>(); list.add(3.4);

 C. ArrayList<Double> list = new ArrayList<>(); list.add(3);

 D. ArrayList<Integer> list = new ArrayList<>(); list.add(3);

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is BD

*Section 11.13 The protected Data and Methods*

***11.53***  What modifier should you use on a class so that a class in the same package can access it but a class in a different package cannot access it?

 A. public

 B. private

 C. protected

 D. Use the default modifier.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***11.54***  What modifier should you use on the members of a class so that they are not accessible to another class in a different package, but are accessible to any subclasses in any package?

 A. public

 B. private

 C. protected

 D. Use the default modifier.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.55***  The visibility of these modifiers increases in this order:

 A. private, protected, none (if no modifier is used), and public.

 B. private, none (if no modifier is used), protected, and public.

 C. none (if no modifier is used), private, protected, and public.

 D. none (if no modifier is used), protected, private, and public.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.56***  A class design requires that a particular member variable must be accessible by any subclasses of this class, but otherwise not by classes which are not members of the same package. What should be done to achieve this?

 A. The variable should be marked public.

 B. The variable should be marked private.

 C. The variable should be marked protected.

 D. The variable should have no special access modifier.

 E. The variable should be marked private and an accessor method provided.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C  
Explanation: See the section on the protected modifier.

***11.57***  Which of the following statements is false?

 A. A public class can be accessed by a class from a different package.

 B. A private method cannot be accessed by a class in a different package.

 C. A protected method can be accessed by a subclass in a different package.

 D. A method with no visibility modifier can be accessed by a class in a different package.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

***11.58***  Which statements are most accurate regarding the following classes?  
  
class A {  
  private int i;  
  protected int j;  
  
  public int getI() {  
    return i;  
  }  
  
  public int getJ() {  
    return j;  
  }  
}  
  
class B extends A {  
  private int k;  
  protected int m;  
  
  public int getK() {  
    return k;  
  }  
  
  public int getM() {  
    return m;  
  }  
}

 A. An object of B contains data fields i, j, k, m.

 B. An object of B contains data fields j, k, m.

 C. An object of B contains data fields j, m.

 D. An object of B contains data fields k, m.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***11.59***  Which statements are most accurate regarding the following classes?  
  
class A {  
  private int i;  
  protected int j;  
}  
  
class B extends A {  
  private int k;  
  protected int m;  
  
  // some methods omitted  
}

 A. In the class B, an instance method can only access i, j, k, m.

 B. In the class B, an instance method can only access j, k, m.

 C. In the class B, an instance method can only access j, m.

 D. In the class B, an instance method can only access k, m.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

*Section Comprehensive*

***11.61***  Polymorphism means \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

 A. that data fields should be declared private.

 B. that a class can extend another class.

 C. that a variable of supertype can refer to a subtype object.

 D. that a class can contain another class.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.62***  Encapsulation means \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

 A. that data fields should be declared private.

 B. that a class can extend another class.

 C. that a variable of supertype can refer to a subtype object.

 D. that a class can contain another class.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***11.63***  Inheritance means \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

 A. that data fields should be declared private.

 B. that a class can extend another class.

 C. that a variable of supertype can refer to a subtype object.

 D. that a class can contain another class.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***11.64***  Composition means \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

 A. that data fields should be declared private.

 B. that a class extends another class.

 C. that a variable of supertype refers to a subtype object.

 D. that a class contains a data field that references another object.

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D

*Section 14.3 Exception Types*

***14.1***  A Java exception is an instance of \_\_\_\_\_\_\_\_\_\_.

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. NumberFormatException

***14.2***  An instance of \_\_\_\_\_\_\_\_\_ describes system errors. If this type of error occurs, there is little you can do beyond notifying the user and trying to terminate the program gracefully.

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. NumberFormatException

***14.3***  An instance of \_\_\_\_\_\_\_\_\_ describes the errors caused by your program and external circumstances. These errors can be caught and handled by your program.

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. NumberFormatException

***14.4***  An instance of \_\_\_\_\_\_\_\_\_ describes programming errors, such as bad casting, accessing an out-of-bounds array, and numeric errors..

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. NumberFormatException

***14.5***  The following code causes Java to throw \_\_\_\_\_\_\_\_\_.  
int number = Integer.MAX\_VALUE + 1;

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. no exceptions

***14.6***  An instance of \_\_\_\_\_\_\_\_\_ are unchecked exceptions.

 A. RuntimeException

 B. Exception

 C. Error

 D. Throwable

 E. NumberFormatException

***14.7***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    System.out.println(1 / 0);  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. ClassCastException

 E. No exception

***14.8***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    int[] list = new int[5];  
    System.out.println(list[5]);  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. ClassCastException

 E. No exception

***14.9***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    String s = "abc";  
    System.out.println(s.charAt(3));  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. ClassCastException

 E. No exception

***14.10***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    Object o = new Object();  
    String d = (String)o;  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. ClassCastException

 E. No exception

***14.11***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    Object o = null;  
    System.out.println(o.toString());  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. ClassCastException

 E. NullPointerException

***14.12***  What exception type does the following program throw?  
public class Test {  
  public static void main(String[] args) {  
    Object o = null;  
    System.out.println(o);  
  }  
}

 A. ArithmeticException

 B. ArrayIndexOutOfBoundsException

 C. StringIndexOutOfBoundsException

 D. No exception

 E. NullPointerException

*Section 14.4 More on Exception Handling*

***14.13***  A method must declare to throw \_\_\_\_\_\_\_\_.

 A. unchecked exceptions

 B. checked exceptions

 C. Error

 D. RuntimeException

***14.14***  Which of the following statements are true?

 A. You use the keyword throws to declare exceptions in the method heading.

 B. A method may declare to throw multiple exceptions.

 C. To throw an exception, use the key word throw.

 D. If a checked exception occurs in a method, it must be either caught or declared to be thrown from the method.

***14.15***  Analyze the following code:  
  
class Test {  
  public static void main(String[] args)  
    throws MyException {  
    System.out.println("Welcome to Java");  
  }  
}  
  
class MyException extends Error {  
}

 A. You should not declare a class that extends Error, because Error raises a fatal error that terminates the program.

 B. You cannot declare an exception in the main method.

 C. You declared an exception in the main method, but you did not throw it.

 D. The program has a compilation error.

***14.16***  Analyze the following code:  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      String s = "5.6";  
      Integer.parseInt(s); // Cause a NumberFormatException  
  
      int i = 0;  
      int y = 2 / i;  
    }  
    catch (Exception ex) {  
      System.out.println("NumberFormatException");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("RuntimeException");  
    }  
  }  
}

 A. The program displays NumberFormatException.

 B. The program displays RuntimeException.

 C. The program displays NumberFormatException followed by RuntimeException.

 D. The program has a compilation error.

***14.17***  Analyze the following program.  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      String s = "5.6";  
      Integer.parseInt(s); // Cause a NumberFormatException  
  
      int i = 0;  
      int y = 2 / i;  
      System.out.println("Welcome to Java");  
    }  
    catch (Exception ex) {  
      System.out.println(ex);  
    }  
  }  
}

 A. An exception is raised due to Integer.parseInt(s);

 B. An exception is raised due to 2 / i;

 C. The program has a compilation error.

 D. The program compiles and runs without exceptions.

***14.18***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      method();  
      System.out.println("After the method call");  
    }  
    catch (NumberFormatException ex) {  
      System.out.println("NumberFormatException");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("RuntimeException");  
    }  
  }  
  
  static void method() {  
    String s = "5.6";  
    Integer.parseInt(s); // Cause a NumberFormatException  
  
    int i = 0;  
    int y = 2 / i;  
    System.out.println("Welcome to Java");  
  }  
}

 A. The program displays NumberFormatException.

 B. The program displays NumberFormatException followed by After the method call.

 C. The program displays NumberFormatException followed by RuntimeException.

 D. The program has a compilation error.

 E. The program displays RuntimeException.

***14.19***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      method();  
      System.out.println("After the method call");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("RuntimeException");  
    }  
    catch (Exception ex) {  
      System.out.println("Exception");  
    }  
  }  
  
  static void method() throws Exception {  
    try {  
      String s = "5.6";  
      Integer.parseInt(s); // Cause a NumberFormatException  
  
      int i = 0;  
      int y = 2 / i;  
      System.out.println("Welcome to Java");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("RuntimeException");  
    }  
    catch (Exception ex) {  
      System.out.println("Exception");  
    }  
  }  
}

 A. The program displays RuntimeException twice.

 B. The program displays Exception twice.

 C. The program displays RuntimeException followed by After the method call.

 D. The program displays Exception followed by RuntimeException.

 E. The program has a compilation error.

*Section 14.5 The finally Clause*

***14.20***  What is wrong in the following program?  
  
class Test {  
  public static void main (String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
     }  
  }  
}

 A. You cannot have a try block without a catch block.

 B. You cannot have a try block without a catch block or a finally block.

 C. A method call that does not declare exceptions cannot be placed inside a try block.

 D. Nothing is wrong.

***14.21***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main (String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
    }  
    finally {  
      System.out.println("The finally clause is executed");  
    }  
  }  
}

 A. Welcome to Java

 B. Welcome to Java followed by The finally clause is executed in the next line

 C. The finally clause is executed

 D. None of the above

***14.22***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main (String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      return;  
    }  
    finally {  
      System.out.println("The finally clause is executed");  
    }  
  }  
}

 A. Welcome to Java

 B. Welcome to Java followed by The finally clause is executed in the next line

 C. The finally clause is executed

 D. None of the above

***14.23***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      int i = 0;  
      int y = 2 / i;  
      System.out.println("Welcome to HTML");  
    }  
    finally {  
      System.out.println("The finally clause is executed");  
    }  
  }  
}

 A. Welcome to Java, then an error message.

 B. Welcome to Java followed by The finally clause is executed in the next line, then an error message.

 C. The program displays three lines: Welcome to Java, Welcome to HTML, The finally clause is executed, then an error message.

 D. None of the above.

***14.24***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      int i = 0;  
      double y = 2.0 / i;  
      System.out.println("Welcome to HTML");  
    }  
    finally {  
      System.out.println("The finally clause is executed");  
    }  
  }  
}

 A. Welcome to Java.

 B. Welcome to Java followed by The finally clause is executed in the next line.

 C. The program displays three lines: Welcome to Java, Welcome to HTML, The finally clause is executed.

 D. None of the above.

***14.25***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      int i = 0;  
      int y = 2/i;  
      System.out.println("Welcome to Java");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("Welcome to Java");  
    }  
    finally {  
      System.out.println("End of the block");  
    }  
  }  
}

 A. The program displays Welcome to Java three times followed by End of the block.

 B. The program displays Welcome to Java two times followed by End of the block.

 C. The program displays Welcome to Java three times.

 D. The program displays Welcome to Java two times.

***14.26***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      int i = 0;  
      int y = 2/i;  
      System.out.println("Welcome to Java");  
    }  
    catch (RuntimeException ex) {  
      System.out.println("Welcome to Java");  
    }  
    finally {  
      System.out.println("End of the block");  
    }  
     
    System.out.println("End of the block");  
  }  
}

 A. The program displays Welcome to Java three times followed by End of the block.

 B. The program displays Welcome to Java two times followed by End of the block.

 C. The program displays Welcome to Java two times followed by End of the block two times.

 D. You cannot catch RuntimeException errors.

***14.27***  What is displayed on the console when running the following program?  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      System.out.println("Welcome to Java");  
      int i = 0;  
      int y = 2/i;  
      System.out.println("Welcome to Java");  
    }  
    finally {  
      System.out.println("End of the block");  
    }  
     
    System.out.println("End of the block");  
  }  
}

 A. The program displays Welcome to Java three times followed by End of the block.

 B. The program displays Welcome to Java two times followed by End of the block.

 C. The program displays Welcome to Java two times followed by End of the block two times.

 D. The program displays Welcome to Java and End of the block, and then terminates because of an unhandled exception.

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

Click here to show the correct answer

*Section 14.6 When to Use Exceptions*

***14.28***  Which of the following is not an advantage of Java exception handling?

 A. Java separates exception handling from normal processing tasks.

 B. Exception handling improves performance.

 C. Exception handling makes it possible for the caller's caller to handle the exception.

 D. Exception handling simplifies programming because the error-reporting and error-handling code can be placed at the catch block.

***14.29***  Analyze the following code:  
  
class Test {  
  public static void main(String[] args) {  
    try {  
      int zero = 0;  
      int y = 2/zero;  
      try {  
        String s = "5.6";  
        Integer.parseInt(s); // Cause a NumberFormatException  
      }  
      catch(Exception e) {  
      }  
    }  
    catch(RuntimeException e) {  
      System.out.println(e);  
    }  
  }  
}

 A. A try-catch block cannot be embedded inside another try-catch block.

 B. A good programming practice is to avoid nesting try-catch blocks, because nesting makes programs difficult to read. You can rewrite the program using only one try-catch block.

 C. The program has a compilation error because Exception appears before RuntimeException.

 D. None of the above.

*Section 14.10 The File Class*

***14.31***  What are the reasons to create an instance of the File class?

 A. To determine whether the file exists.

 B. To obtain the properties of the file such as whether the file can be read, written, or is hidden.

 C. To rename the file.

 D. To delete the file.

 E. To read/write data from/to a file

***14.32***  Which of the following returns the path separator character?

 A. File.pathSeparator

 B. File.pathSeparatorChar

 C. File.separator

 D. File.separatorChar

 E. None of the above.

***14.33***  Which of the following statements creates an instance of File on Window for the file c:\temp.txt?

 A. new File("c:\temp.txt")

 B. new File("c:\\temp.txt")

 C. new File("c:/temp.txt")

 D. new File("c://temp.txt")

***14.34***  Which of the following statements are true?

 A. If a file (e.g., c:\temp.txt) does not exist, new File("c:\\temp.txt") returns null.

 B. If a directory (e.g., c:\liang) does not exist, new File("c:\liang") returns null.

 C. If a file (e.g., c:\temp.txt) does not exist, new File("c:\\temp.txt") creates a new file named c:\temp.txt.

 D. If a directory (e.g., c:\liang) does not exist, new File("c:\liang") creates a new directory named c:\liang.

 E. None of the above.

*Section 14.11 Text I/O*

***14.35***  Which class contains the method for checking whether a file exists?

 A. File

 B. PrintWriter

 C. Scanner

 D. System

***14.36***  Which class do you use to write data into a text file?

 A. File

 B. PrintWriter

 C. Scanner

 D. System

Indicate which book and edition you are using. Thanks!

***14.37***  Which class do you use to read data from a text file?

 A. File

 B. PrintWriter

 C. Scanner

 D. System

***14.38***  Which method can be used to write data?

 A. close

 B. print

 C. exist

 D. rename

***14.39***  Which method can be used to read a whole line from the file?

 A. next

 B. nextLine

 C. nextInt

 D. nextDouble

***14.40***  Which method can be used to create an input object for file temp.txt?

 A. new Scanner("temp.txt")

 B. new Scanner(temp.txt)

 C. new Scanner(new File("temp.txt"))

 D. new Scanner(File("temp.txt"))

***14.41***  Suppose you enter 34.3 57.8 789, then press the ENTER key. Analyze the following code.  
Scanner input = new Scanner(System.in);  
int v1 = input.nextInt();  
int v2 = input.nextInt();  
String line = input.nextLine();

 A. After the last statement is executed, v1 is 34.

 B. The program has a runtime error because 34.3 is not an integer.

 C. After the last statement is executed, line contains characters '7', '8', '9', '\n'.

 D. After the last statement is executed, line contains characters '7', '8', '9'.

***14.42***  Suppose you enter 34.3 57.8 789, then press the ENTER key. Analyze the following code.  
Scanner input = new Scanner(System.in);  
double v1 = input.nextDouble();  
double v2 = input.nextDouble();  
String line = input.nextLine();

 A. After the last statement is executed, line contains characters '7', '8', '9'.

 B. After the last statement is executed, line contains characters '7', '8', '9', '\n'.

 C. After the last statement is executed, line contains characters ' ', '7', '8', '9', '\n'.

 D. After the last statement is executed, line contains characters ' ', '7', '8', '9'.

***14.43***  Suppose you enter 34.3, the ENTER key, 57.8, the ENTER key. Analyze the following code.  
  1 Scanner input = new Scanner(System.in);  
  2 double v1 = input.nextDouble();  
  3 double v2 = input.nextDouble();  
  4 String line = input.nextLine();

 A. After line 2 is executed, v1 is 34.3.

 B. After line 3 is executed, v2 is 57.8.

 C. After line 4 is executed, line contains an empty string.

 D. After line 4 is executed, line is null.

 E. After line 4 is executed, line contains character "\n".

***14.44***  Suppose you enter 34.3, the ENTER key, 57.8, the ENTER key, abc, the Enter key. Analyze the following code.  
  1 Scanner input = new Scanner(System.in);  
  2 double v1 = input.nextDouble();  
  3 double v2 = input.nextDouble();  
  4 String line = input.nextLine();

 A. After line 2 is executed, v1 is 34.3.

 B. After line 3 is executed, v2 is 57.8.

 C. After line 4 is executed, line contains an empty string.

 D. After line 4 is executed, line is null.

 E. After line 4 is executed, line contains character "abc".

***14.45***  Which method can be used to create an output object for file temp.txt?

 A. new PrintWriter("temp.txt")

 B. new PrintWriter(temp.txt)

 C. new PrintWriter(new File("temp.txt"))

 D. new PrintWriter(File("temp.txt"))

*Section 14.13 Reading Data from the Web*

***14.46***  To create an InputStream to read from a file on a Web server, you use the method \_\_\_\_\_\_\_\_\_\_ in the URL class.

 A. getInputStream();

 B. obtainInputStream();

 C. openStream();

 D. connectStream();

*Section 15.2 Abstract Classes*

***15.1***  Which of the following class definitions defines a legal abstract class?

 A. class A { abstract void unfinished() { } }

 B. class A { abstract void unfinished(); }

 C. abstract class A { abstract void unfinished(); }

 D. public class abstract A { abstract void unfinished(); }

***15.2***  Which of the following declares an abstract method in an abstract Java class?

 A. public abstract method();

 B. public abstract void method();

 C. public void abstract Method();

 D. public void method() {}

 E. public abstract void method() {}

***15.3***  Which of the following statements regarding abstract methods are true?

 A. An abstract class can have instances created using the constructor of the abstract class.

 B. An abstract class can be extended.

 C. A subclass of a non-abstract superclass can be abstract.

 D. A subclass can override a concrete method in a superclass to declare it abstract.

 E. An abstract class can be used as a data type.

***15.4***  Which of the following statements regarding abstract methods are true?

 A. Abstract classes have constructors.

 B. A class that contains abstract methods must be abstract.

 C. It is possible to declare an abstract class that contains no abstract methods.

 D. An abstract method cannot be contained in a nonabstract class.

 E. A data field can be declared abstract.

***15.5***  Suppose A is an abstract class, B is a concrete subclass of A, and both A and B have a no-arg constructor. Which of the following is correct?

 A. A a = new A();

 B. A a = new B();

 C. B b = new A();

 D. B b = new B();

***15.6***  What is the output of running class Test?  
  
public class Test {  
  public static void main(String[] args) {  
    new Circle9();  
  }  
}  
  
public abstract class GeometricObject {  
  protected GeometricObject() {  
    System.out.print("A");  
  }  
  
  protected GeometricObject(String color, boolean filled) {  
    System.out.print("B");  
  }  
}  
  
public class Circle9 extends GeometricObject {  
  /\*\* Default constructor \*/  
  public Circle9() {  
    this(1.0);  
    System.out.print("C");  
  }  
  
  /\*\* Construct circle with a specified radius \*/  
  public Circle9(double radius) {  
    this(radius, "white", false);  
    System.out.print("D");  
  }  
  
  /\*\* Construct a circle with specified radius, filled, and color \*/  
  public Circle9(double radius, String color, boolean filled) {  
    super(color, filled);  
    System.out.print("E");  
  }  
}

 A. ABCD

 B. BACD

 C. CBAE

 D. AEDC

 E. BEDC

*Section 15.3 Case Study: the Abstract Number Class*

***15.7***  The java.lang.Number and its subclasses are introduced in Chapter 11. Analyze the following code.  
  
    Number numberRef = new Integer(0);  
    Double doubleRef = (Double)numberRef;

 A. There is no such class named Integer. You should use the class Int.

 B. The compiler detects that numberRef is not an instance of Double.

 C. A runtime class casting exception occurs, since numberRef is not an instance of Double.

 D. The program runs fine, since Integer is a subclass of Double.

 E. You can convert an int to double, so you can cast an Integer instance to a Double instance.

***15.8***  Analyze the following code.  
  
    Number[] numberArray = new Integer[2];  
    numberArray[0] = new Double(1.5);

 A. You cannot use Number as a data type since it is an abstract class.

 B. Since each element of numberArray is of the Number type, you cannot assign an Integer object to it.

 C. Since each element of numberArray is of the Number type, you cannot assign a Double object to it.

 D. At runtime, new Integer[2] is assigned to numberArray. This makes each element of numberArray an Integer object. So you cannot assign a Double object to it.

***15.9***  Analyze the following code.  
public class Test {  
  public static void main(String[] args) {  
    Number x = new Integer(3);  
    System.out.println(x.intValue());  
    System.out.println(x.compareTo(new Integer(4)));  
  }  
}

 A. The program has a compile error because an Integer instance cannot be assigned to a Number variable.

 B. The program has a compile error because intValue is an abstract method in Number.

 C. The program has a compile error because x does not have the compareTo method.

 D. The program compiles and runs fine.

***15.10***  Analyze the following code.  
public class Test {  
  public static void main(String[] args) {  
    Number x = new Integer(3);  
    System.out.println(x.intValue());  
    System.out.println((Integer)x.compareTo(new Integer(4)));  
  }  
}

 A. The program has a compile error because an Integer instance cannot be assigned to a Number variable.

 B. The program has a compile error because intValue is an abstract method in Number.

 C. The program has a compile error because x cannot be cast into Integer.

 D. The program has a compile error because the member access operator (.) is executed before the casting operator.

 E. The program compiles and runs fine.

***15.11***  Which of the following statements are correct?

 A. Integer i = 4.5;

 B. Double i = 4.5;

 C. Object i = 4.5;

 D. Number i = 4.5;

*Section 15.4 Case Study: Calendar and GregorianCalendar*

***15.12***  The java.util.Calendar and java.util.GregorianCalendar classes are introduced in Chapter 11. Analyze the following code.  
  
1. import java.util.\*;  
2. public class Test {  
3.   public static void main(String[] args) {  
4.     Calendar[] calendars = new Calendar[10];  
5.     calendars[0] = new Calendar();  
6.     calendars[1] = new GregorianCalendar();  
7.   }  
8. }

 A. The program has a compile error on Line 4 because java.util.Calendar is an abstract class.

 B. The program has a compile error on Line 5 because java.util.Calendar is an abstract class.

 C. The program has a compile error on Line 6 because Calendar[1] is not of a GregorianCalendar type.

 D. The program has no compile errors.

***15.13***  Assume Calendar calendar = new GregorianCalendar(). \_\_\_\_\_\_\_\_\_\_ returns the month of the year.

 A. calendar.get(Calendar.MONTH)

 B. calendar.get(Calendar.MONTH\_OF\_YEAR)

 C. calendar.get(Calendar.WEEK\_OF\_MONTH)

 D. calendar.get(Calendar.WEEK\_OF\_YEAR)

***15.14***  Assume Calendar calendar = new GregorianCalendar(). \_\_\_\_\_\_\_\_\_\_ returns the week of the year.

 A. calendar.get(Calendar.MONTH)

 B. calendar.get(Calendar.MONTH\_OF\_YEAR)

 C. calendar.get(Calendar.WEEK\_OF\_MONTH)

 D. calendar.get(Calendar.WEEK\_OF\_YEAR)

***15.15***  Assume Calendar calendar = new GregorianCalendar(). \_\_\_\_\_\_\_\_\_\_ returns the number of days in a month.

 A. calendar.get(Calendar.MONTH)

 B. calendar.get(Calendar.MONTH\_OF\_YEAR)

 C. calendar.get(Calendar.WEEK\_OF\_MONTH)

 D. calendar.get(Calendar.WEEK\_OF\_YEAR)

 E.calendar.getActualMaximum(Calendar.DAY\_OF\_MONTH)

*Section 15.5 Interfaces*

***15.16***  Which of the following is a correct interface?

 A. interface A { void print() { }; }

 B. abstract interface A { print(); }

 C. abstract interface A { abstract void print() { };}

 D. interface A { void print();}

***15.17***  Which of the following is incorrect?

 A. An abstract class contains constructors.

 B. The constructors in an abstract class should be protected.

 C. The constructors in an abstract class are private.

 D. You may declare a final abstract class.

 E. An interface may contain constructors.

***15.18***  \_\_\_\_\_\_\_ is a reference type.

 A. A class type

 B. An interface type

 C. An array type

 D. A primitive type

***15.19***  Show the output of running the class Test in the following code lines:  
  
interface A {  
}  
  
class C {    
}  
  
class B extends D implements A {  
}  
  
public class Test {  
  public static void main(String[] args) {  
    B b = new B();  
    if (b instanceof A)  
      System.out.println("b is an instance of A");  
    if (b instanceof C)  
      System.out.println("b is an instance of C");  
  }  
}  
  
class D extends C {    
}

 A. Nothing.

 B. b is an instance of A.

 C. b is an instance of C.

 D. b is an instance of A followed by b is an instance of C.

***15.20***  Suppose A is an interface, B is a concrete class with a default constructor that implements A. Which of the following is correct?

 A. A a = new A();

 B. A a = new B();

 C. B b = new A();

 D. B b = new B();

*Section 15.6 The Comparable Interface*

***15.21***  Analyze the following code:  
  
public class Test1  {  
  public Object max(Object o1, Object o2) {  
    if ((Comparable)o1.compareTo(o2) >= 0) {  
      return o1;  
    }  
    else {  
      return o2;  
    }  
  }  
}

 A. The program has a compile error because Test1 does not have a main method.

 B. The program has a compile error because o1 is an Object instance and it does not have the compareTo method.

 C. The program has a compile error because you cannot cast an Object instance o1 into Comparable.

 D. The program would compile if ((Comparable)o1.compareTo(o2) >= 0) is replaced by (((Comparable)o1).compareTo(o2) >= 0).

***15.22***  Which of the following statements are true?

 A. The String class implements Comparable<String>.

 B. The Date class implements Comparable<Date>.

 C. The Double class implements Comparable<Double>.

 D. The BigInteger class implements Comparable<BigInteger>.

***15.23***  Analyze the following code.  
  
1. public class Test  {  
2.   public static void main(String[] args) {  
3.     Fruit[] fruits = {new Fruit(2), new Fruit(3), new Fruit(1)};  
4.     java.util.Arrays.sort(fruits);  
5.   }  
6. }  
  
class Fruit {  
  private double weight;  
    
  public Fruit(double weight) {  
    this.weight = weight;  
  }  
}

 A. The program has a compile error because the Fruit class does not have a default constructor.

 B. The program has a runtime error on Line 3 because the Fruit class does not have a default constructor.

 C. The program has a compile error on Line 4 because the Fruit class does not implement the java.lang.Comparable interface and the Fruit objects are not comparable.

 D. The program has a runtime error on Line 4 because the Fruit class does not implement the java.lang.Comparable interface and the Fruit objects are not comparable.

*Section 15.7 The Cloneable Interface*

***15.24***  Analyze the following code.  
  
public class Test {  
  public static void main(String[] args) {  
    java.util.Date x = new java.util.Date();  
    java.util.Date y = x.clone();  
    System.out.println(x = y);  
  }  
}

 A. A java.util.Date object is not cloneable.

 B. x = y in System.out.println(x = y) causes a compile error because you cannot have an assignment statement inside a statement.

 C. x = y in System.out.println(x = y) causes a runtime error because you cannot have an assignment statement inside a statement.

 D. The program has a compile error because the return type of the clone() method is java.lang.Object.

***15.25***  The output from the following code is \_\_\_\_\_\_\_\_\_\_.  
  
java.util.ArrayList<String> list = new java.util.ArrayList<>();  
list.add("New York");   
java.util.ArrayList<String> list1 = (java.util.ArrayList<String>)(list.clone());  
list.add("Atlanta");   
list1.add("Dallas");   
System.out.println(list1);

 A. [New York]

 B. [New York, Atlanta]

 C. [New York, Atlanta, Dallas]

 D. [New York, Dallas]

***15.26***  The GeometricObject and Circle classes are defined in this chapter. Analyze the following code.  
  
public class Test {  
  public static void main(String[] args) {  
    GeometricObject x = new Circle(3);  
    GeometricObject y = (Circle)(x.clone());  
    System.out.println(x);  
    System.out.println(y);  
  }  
}

 A. The program has a compile error because the clone() method is protected in the Object class.

 B. After you override the clone() method and make it public in the Circle class, the problem can compile and run just fine, but y is null if Circle does not implement the Cloneable interface.

 C. To enable a Circle object to be cloned, the Circle class has to override the clone() method and implement the java.lang.Cloneable interface.

 D. If GeometricObject implements Cloneable and Circle overrides the clone() method, the clone() method will work fine to clone Circle objects.

*Section 15.8 Interfaces vs. Abstract Classes*

***15.27***  Which of the following statements are true?

 A. If you compile an interface without errors, a .class file is created for the interface.

 B. If you compile a class without errors but with warnings, a .class file is created.

 C. If you compile a class with errors, a .class file is created for the class.

 D. If you compile an interface without errors, but with warnings, a .class file is created for the interface.

***15.28***  Which of the following statements are true?

 A. Inheritance models the is-a relationship between two classes.

 B. A strong is-a relationship describes a direct inheritance relationship between two classes.

 C. A weak is-a relationship describes that a class has certain properties.

 D. A strong is-a relationship can be represented using class inheritance.

 E. A weak is-a relationship can be represented using interfaces.

***15.29***  What is the best suitable relationship between Employee and Faculty?

 A. Composition

 B. Aggregation

 C. Inheritance

 D. None.

***15.30***  Assume an employee can work for only one company. What is the best suitable relationship between Company and Employee?

 A. None

 B. Aggregation

 C. Inheritance

 D. Composition

***15.31***  The relationship between an interface and the class that implements it is

 A. Composition

 B. Aggregation

 C. Inheritance

 D. None

*Section 15.9 Case Study: The Rational Class*

***15.32***  The Rational class in this chapter is defined as a subclass of java.lang.Number. Which of the following expressions is correct?

 A. Rational.doubleValue();

 B. Rational.doubleValue("5/4");

 C. new Rational(5, 4).doubleValue();

 D. new Rational(5, 4).toDoubleValue();

 E. new Rational(5, 4).intValue();

***15.33***  The Rational class in this chapter extends java.lang.Number and implements java.lang.Comparable. Analyze the following code.  
  
1. public class Test {  
2.   public static void main(String[] args) {  
3.     Number[] numbers = {new Rational(1, 2), new Integer(4), new Double(5.6)};  
4.     java.util.Arrays.sort(numbers);  
5.   }  
6. } 

 A. The program has a compile error because numbers is declared as Number[], so you cannot assign {new Rational(1, 2), new Integer(4), new Double(5.6)} to it.

 B. The program has a runtime error because numbers is declared as Number[], so you cannot assign {new Rational(1, 2), new Integer(4), new Double(5.6)} to it.

 C. The program has a compile error because numbers is declared as Number[], so you cannot pass it to Arrays.sort(Object[]).

 D. The program has a runtime error because the compareTo methods in Rational, Integer, and Double classes do not compare the value of one type with a value of another type.