Chapter 3: Using Classes and Objects

Multiple Choice Questions:

1) The _____ operator is used to instantiate an object.

a) `static`
b) `new`
c) `+`
d) `-`
e) none of the above

Answer: b

Explanation: The `new` operator instantiates an object. There is no `static` operator, and the `+` and `-` operators are for arithmetic expressions (although the + operator can also be used for String concatenation).

2) A special method that is invoked to set up an object during instantiation is called a _____.

a) new method
b) dot operator
c) creator
d) constructor
e) destructor

Answer: d

Explanation: The constructor is called to set up an object during instantiation. The dot operator is used to access methods of an object. There is no "new" method – new is an operator. Java also does not have "creators" or "destructors."

3) Which of the following is an *invalid* way to instantiate a `String` object?

a) `String title = new String("Java Software Solutions");`
b) `String name = "John Lewis";`
c) `String empty = "";`
d) `String alsoEmpty = new String("");`
e) all of the above are valid

Answer: e

Explanation: Choices a and d represent the standard approach to instantiating a `String` object. Choice d creates the emptry string. Choices b and c use a shortcut notation that is only available for creating a `String` object.

4) Assume that we have a `Random` object referenced by a variable called `generator`. Which of the following lines will generate a random number in the range 5-20 and store it in the `int` variable `randNum`?

a) `randNum = generator.nextInt(15) + 5;`

b) `randNum = generator·nextInt(15) + 6;`

c) `randNum = generator·nextInt(16) + 5;`

d) `randNum = generator·nextInt(16) + 6;`

e) none of the above

Answer: c

Explanation: When called with *16* as a parameter, the *nextInt()* method will return a number in the range 0 to 15. Adding 5 to this random number will generate a number in the range 5 to 20.

5) Which of the following classes include the *getCurrencyInstance()* method?

a) *String*

b) *NumberFormat*

c) *DecimalFormat*

d) *Math*

e) none of the above

Answer: b

Explanation: The *NumberFormat* class includes the *getCurrencyInstance()* method.

6) Which of the following expressions correctly compute $5 + 2^6$?

a) `result = 5 + 2^6;`

b) `result = 5 + 2*exponent(6);`

c) `result = 5 + 2*Math·exponent(6);`

d) `result = 5 + Math·pow(2, 6);`

e) none of the above

Answer: d

Explanation: Choice a is wrong because Java does not have an exponential operator for primitive types. Choices b and C are wrong because there are no methods named exponent in the *java·lang* package or the *Math* class. Choice c correctly uses the *Math·pow* method to compute the expression.

7) Consider the following snippet of code:

```
Random generator = new Random();
int randNum = generator·nextInt(20) + 1;
```

Which of the following will be true after these lines are executed?

a) *randNum* will hold a number between 1 and 20 inclusive.

b) *randNum* will hold a number between 0 and 20 inclusive.

c) *randNum* will hold a number between 1 and 21 inclusive.
d) these lines will not be executed because a compiler error will result.
e) none of the above

Answer: a

Explanation: When called with a parameter of *20*, the *nextInt()* method will return an integer between 0 and 19 inclusive. Adding one to this will result in a number between 1 and 20 inclusive.

8) Which of the following represents the proper way to create a *NumberFormat* object that formats numbers as percentages?

a) `NumberFormat fmt = new NumberFormat(%);`
b) `NumberFormat fmt = new NumberFormat("%");`
c) `NumberFormat fmt = NumberFormat·getPercentInstance();`
d) `NumberFormat fmt = new PercentNumberFormat();`
e) none of the above

Answer: c

Explanation: The *NumberFormat* class uses factory methods to construct objects. The new operator is called implicitly in this case. The factory method that is used is called *getPercentInstance()*. Therefore choice c is correct. Choice a and Choice b are incorrect since they call the new operator explicitly. Choice d is incorrect because there is no *PercentNumberFormat* class that can be called with the new constructor.

9) Which of the following is a correct declaration of enumerated type for the suits of a deck of cards?

a) `enumerated type Suit = { hearts, spades, diamonds, clubs };`
b) `enum Suit {hearts, spades, diamonds, clubs };`
c) `enum Suit {hearts, spades, diamonds, clubs }`
d) `enumerated type Suit = {hearts, spades, diamonds, clubs };`
e) `enum Suit = { hearts, spades, diamonds, clubs }`

Answer: c

Explanation: Choice c represents the correct syntax for declaring an enumerated type called Suit that can take one of the values hearts, spades, diamonds or clubs.

10) _____ is the automatic conversion between a primitive value and a corresponding wrapper object.

a) Generating
b) Aliasing
c) Number formatting
d) Static invocation
e) Autoboxing

Answer: e

Explanation: Autoboxing allows a programmer to automatically convert between a primitive value and a

corresponding wrapper object. The other 4 answers are incorrect.

11) Which of the following best describes what happens when an object no longer has any references pointing to it?

a) The object is overwritten the next time the new operator is called.
b) The object is overwritten the next time the new operator is called using the same class.
c) The object is immediately deleted from memory.
d) The object is marked as garbage and its associated memory is freed when the garbage collector runs.
e) The object stays in memory for the remainder of the programs execution.

Answer: d
Explanation: Java has automatic garbage collection. When an object no longer has any references pointing to it, it is marked as garbage. When the garbage collector runs, the memory is freed so that new objects can be created in its space.

12) When an object variable is declared but it is not assigned a value, it is called a _____.

a) null reference
b) empty reference
c) void reference
d) zero reference
e) static reference

Answer: a
Explanation: A null reference is a reference that does not refer to any object.

13) Suppose we have a *String* object referenced by a variable called *listing*. Suppose we want a new *String* object that consists of the first 5 characters in listing. Which of the following lines of code will achieve this?

a) `String prefix = listing.front(5);`
b) `String prefix = listing.front(6);`
c) `String prefix = listing.substring(1,5);`
d) `String prefix = listing.substring(0,5);`
e) `String prefix = listing.firstChars(5);`

Answer: d
Explanation: Choices a, b, and e are wrong because the *String* class does not have methods called *front* or *firstChars*. Choice c is incorrect because the first character in the string is indexed by 1. Choice c is correct, since it will create a new string from the first 5 characters of *listing*.

14) When two references point to the same object, _____ .

a) a run-time error will occur.
b) a compiler error will occur.
c) the references are called *aliases* of each other.
d) the object will be marked for garbage collection.
e) the references are called *null* references.

Answer: c

Explanation: It is perfectly acceptable to have two references pointing to the same object, so no errors will be generated. Therefore choices a and b are incorrect. Choice d is incorrect since objects are marked for garbage collection only when *no* references point to them. Choice e is incorrect since references are called *null* when they do not point to anything. Choice c is the correct answer.

15) The String class _____ .

a) is part of the java.lang package.
b) is part of the java.util.package.
c) is a wrapper class.
d) none of the above.
e) all of the above.

Answer: a

Explanation: The `String` class is part of the `java.lang` package. Therefore, choice a is the correct answer. It is not a wrapper class, and it is not part of the `java.util` package, therefore the other choices are wrong.

True/False Questions:

1) Multiple reference variables can refer to the same object.
Answer: True
Explanation: When multiple reference variables refer to the same object, they are called *aliases* of each other.

2) The new operator is used to access an objects methods.
Answer: False
Explanation: The new operator is used to instantiate objects. The dot operator is used to access an objects methods.

3) A variable that is declared as a primitive type stores the actual value of the associated data. A variable that is declared as an object type stores a pointer to the associated object.
Answer: True
Explanation: Primitive data types are stored by value, object types are stored by reference.

4) When there are no references to an object, the object is marked as *garbage* and is automatically removed from memory using Java's automatic garbage collection feature.
Answer: True
Explanation: An object may lose all references to it through reassignment. In this case, it serves no useful purpose since its methods and data can no longer be accessed. At this point, Java's automatic garbage collection feature deletes it from memory, freeing up the space for new objects.

5) `String` objects can be changed after instantiation.
Answer: False
Explanation: `String` objects are immutable, meaning that once a `String` object is created, its value cannot be lengthened or shortened, nor can any of its characters change.

6) All of the classes contained in the `java.util` package are automatically included in every Java program.
Answer: False
Explanation: In order to use classes in the `java.util` package, the programmer must include them using an import statement. The classes in the `java.lang` package are automatically included in every Java program, and therefore do not require an import statement to use.

7) When called with integer parameter `n`, the `nextInt()` method of the `Random` class will return a randomly generated integer between 0 and n.
Answer: False
Explanation: When called with integer parameter `n`, the `nextInt()` method of the `Random` class will return a randomly generated integer between 0 and n-1.

8) The `Math` class is part of the `java.lang` package.
Answer: True
Explanation: The `Math` class is part of the `java.lang` package. Therefore, a programmer does not need to include an explicit `import` statement to use its methods.

9) Enumerated types allow a programmer to treat primitive data as objects.

Answer: False

Explanation: Enumerated types are programmer-defined types that allow variables to be assigned values from a specified set. Wrapper classes allow a programmer to treat primitive data as objects.

10) The `System.out.printf()` method is an alternative way to output information in Java.

Answer: True

Explanation: The `System.out.printf()` method provides an alternative way of outputting data, but was mainly incorporated into Java to make it easier to port legacy programs written in C to the Java language.

Short Answer Questions:

1) Explain how variables representing objects and variables representing primitive types are different.

Answer: A variable representing a primitive type actually holds the primitive value associated with the variable, while a variable representing an object holds a reference to the object associated with the variable.

2) Suppose we have a *String* object called *myString*. Write a single line of Java code that will output *myString* in such a way that all of its characters are uppercase.

Answer:

```
System.out.println(myString.toUpperCase());
```

3) Explain what it means for a String object to be *immutable*. Are there any workarounds for this?

Answer: *String* objects are immutable, meaning that once a *String* object is created, its value cannot be lengthened or shortened, nor can any of its characters change. However, there are several methods in the *String* class that return new modified *String* objects. By reassigning the original reference to the result of calling one of these methods, we can effectively change the value of a *String* object.

4) Write a short program that allows the user to input a positive integer and then outputs a randomly generated integer between 1 and the input number.

Answer:

```
import java.util.Scanner;
import java.util.Random;

public class RandomInteger  {
        public static void main(String [] args)  {
                int inputNum, randNum;

                Scanner input = new Scanner(System.in);
                Random generator = new Random();

                System.out.print("Please enter a positive integer: ");
                inputNum = input.nextInt();

                randNum = generator.nextInt(inputNum) + 1;

                System.out.println("Your random number is " + randNum + ".");
```

```
        }//end main
}//end class
```

5) Write a statement that computes the square root of a variable called discriminant and stores the value in a variable called result.

Answer:

```
result = Math.sqrt(discriminant);
```

6) Write a short program that asks the user to input a string, and then outputs the number of characters in the string.

Answer:

```
import java.util.Scanner;

public class StringLength  {
        public static void main(String [] args)  {
                String inputString;

                Scanner input = new Scanner(System.in);

                System.out.print("Please enter a string: ");
                inputString = input.nextLine();

                System.out.println("Your string has " + inputString.length()
                                + " characters!");

        }//end main
}//end class
```

7) Suppose a `Random` object has been created called `generator`. Write a single statement that generates a random number in the range 73 to 100.

Answer:

```
randNum = generator.nextInt(28) + 73;
```

8) What is the range of integers that will be generated by the following expression?

```
generator.nextInt(15) + 5;
```

Answer: This expression will generate a random number in the range 5 to 19 (inclusive).

9) Write an expression that will compute the length of the tangent of a right triangle, given one of the non-right angles. You may assume that this value is stored in a variable called *angle*.

Answer:

```
tangent = Math.tan(angle);
```

10) Write a declaration for an enumerated type that represents the months of the year.

Answer:

```
enum Suit { January, February, March, April, May, June, July, August, September, October, November, December }
```

11) Write a short program that allows the user to enter the base and height of a triangle and outputs the hypotenuse, formatted to three decimal places.

Answer:

```
import java.util.Scanner;
import java.text.DecimalFormat;

public class Hypotenuse  {
        public static void main(String [] args)  {
                double base, height, hypotenuse;

                Scanner input = new Scanner(System.in);
                DecimalFormat fmt = new DecimalFormat("0.###");

                System.out.print("Please enter the base: ");
                base = input.nextDouble();

                System.out.print("Please enter the height: ");
                height = input.nextDouble();

                hypotenuse = Math.sqrt(Math.pow(base,2) + Math.pow(height,2));

                System.out.println("The hypotenuse is " + fmt.format(hypotenuse)
```

```
                + ".");


        }//end main
}//end class
```

12) Write a single statement that creates a DecimalFormat object that formats numbers to 2 decimal places.

Answer:

```
DecimalFormat fmt = new DecimalFormat("0.##");
```

13) What is output by the following code fragment?

```
String hello = new String("Hello World!");
hello = hello.replace('H', 'W');
hello = hello.replace('W', 'H');
System.out.println(hello);
```

Answer: This will output

```
Hello Horld!
```

14) Write a single line that creates a wrapper for an *int* variable *num*.

Answer:

```
Integer numWrapper = new Integer(num);
```

15) Write an expression that computes 12 raised to the power 4.3 and store the result in a double called *result*.

Answer:

```
result = Math.pow(12, 4.3);
```