# SCJA Exercises Day 2

- **Multiple Choice**

In today's lab we give questions covering the following topics:

- Fundamental Data Types
- Object Oriented Concepts
- Object Associations
- Object Abstractions
- Conditional Logic
- Iterative Loops

- **Coding Questions**

  - Coding Question 1 (Fundamental Data Types)
  - Coding Question 2 (Conditional Logic)
  - Coding Question 3 (Loops)

## Multiple Choice

### Fundamental Data Types

1. Which of the following are invalid variable names in Java? (Choose all that apply.)
A. $char
B. 1MyNumber
C. case
D. _int

2. You need to create an application that is used to calculate the attendance at a baseball game. What data type would be most appropriate for storing the attendance?

A. boolean
B. char
C. float
D. int

3. What is the best data type to use if you are required to perform many addition, subtraction, and multiplication calculations on a whole number?

A. float
B. Float
C. int
D. Integer

4. You are writing a class that will store the status of an on/off switch. Which data type is most appropriate to store this value?

A. boolean
B. char
C. float
D. int

5. You have decided on the data type for a variable that will store the information about the on/off switch. Now you must determine a name for it. Which of the following names follows the Java naming conventions?

A. LIGHTSWITCHENABLED
B. LightSwitchEnabled
C. lightSwitchEnabled
D. x

6. What is the best data type to use when storing a status code that may have one of the following values: success, failed, success with errors, undetermined?

A. Object
B. Class
C. boolean
D. enum
E. int

7. Which of the following variables are being set with the use of a literal? (Choose all that apply.)

A. int tvChannel = 4;
B. char c = '5';
C. char d = '\u0123';
D. char e = c;
E. int oldChannel = tvChannel;

8. Consider the following line of code:
short ohMy;
What is the range of values that could be assigned to the variable ohMy?
A. 0 to 2^16 – 1
B. 0 to 2^15 – 1
C. –2^15 – 1 to 2^15 – 1
D. –2^16 – 1 to 2^16 – 1
E. –2^15 to 2^15 – 1
F. –2^15 to 2^15

9. Consider the following line of code:
char ohMy;
What is the range of values that could be assigned to the variable ohMy?
A. 0 to 2^16 – 1
B. 0 to 2^15 – 1
C. –2^15 – 1 to 2^15 – 1
D. –2^16 – 1 to 2^16 – 1
E. –2^15 to 2^15 – 1
F. –2^15 to 2^15

10. Consider the following line of code:
byte ohMy;
What is the range of values that could be assigned to the variable ohMy?
A. 0 to 2^16 – 1
B. 0 to 2^8 – 1
C. –2^7 to 2^7 – 1
D. –2^7 to 2^7
E. –2^15 to 2^15 – 1
F. –2^8 to 2^8 – 1

11. Which of the following statements would not produce a compile error?
A. char my_char = 'c';
B. char your_char = 'int';
C. char what = 'Hello';
D. char what_char = "L";
E. char ok = '\u3456';

12. Which of the following statements are correct?
a) Java has eight primitive types
b) String is a primitive type
c) Variable names can start with upper case letters
d) Local variables must be initialized before they are used

13. When declaring and initializing a variable in a single line of code, what is the correct order?
a) Variable name, assignment, type, semi-colon
b)Assignment, variable name, type, semi-colon
c)Type, assignment, variable name, semi-colon
d)Type, variable name, assignment, semi-colon

14. A signed data type has an equal number of non-zero positive and negative values available.
A. True
B. False

**Answers:**

1. Answer: B and C

B is invalid because it starts with a number, and C is invalid because it is a keyword and thus cannot be a variable name. The first character of an identifier must be a letter, a dollar sign ($), or an underscore (_).

2. Answer:
D. The attendance of a baseball game is going to be a whole number within the range of an int. A, B, and C are incorrect. A is incorrect because boolean variables are used to store literals with values of true or false. B is incorrect because the char data type is used to store a single Unicode character. C is incorrect because float is used to store floating-point numbers.

3. Answer:
C. An int is used to store whole numbers and is a primitive. Primitive variables are faster to perform calculations than their associated wrapper class. A, B, and D are incorrect. A is incorrect because a float is used for floating-point numbers. B is incorrect because Float is a primitive wrapper class used for floating-point numbers. D is incorrect because the Integer data type is the wrapper class for an int. You can tell that it is not a primitive because the first letter is capitalized like all class names. Performing calculations with an Integer would be much slower than the primitive int.

4. Answer:
A. A boolean primitive is used to store true or false which can be applied to a switch. B, C, and D are incorrect. They are all primitives used for different types of data.

5. Answer:
C. A variable should begin with a lowercase letter, with each sequential word capitalized. The name should also be descriptive of what the variable is used for. A, B, and D are incorrect.

6. Answer:
D.An enum or enumeration is used to store data that has the possibility to be one of a few predefined data types. A, B, C, and E are incorrect. A is incorrect because objects are used to store complex data structures. B is incorrect because classes are used to create objects. C and E are incorrect because both are primitives and not suitable for this specific application.

7. Answer:
A, B, and C are correct. A literal is a value that is not a variable. A has the literal 4. B has the literal '5'. C has the literal '\u0123'. D and E are incorrect. D is incorrect because the variable c is used to set this char. E is incorrect since tvChannel is a variable.

8. Answer: E
The data type short is signed and 16 bits in size.

9. Answer: A
The data type char is unsigned and 16 bits in size.

10. Answer: C
The data type byte is signed and 8 bits in size.

11. Answer: A and E
B, C, and D will have compiler errors because a char literal is represented by a single character enclosed in single quotes unless it's being represented as a Unicode value

12. Answer: A,C,D.

13.Answer: D

14. Answer: B

**Object Oriented Programming**

1. How many objects are referenced in this code segment?

```
int numberOfTrees = 5;
Integer ageOfFarm = 14;
float averageHeightOfTrees = 124.2f
Tree treeType;
int[] heightOfEachTree;
```

A. 0
B. 1
C. 2
D. 3
E. 4
F. 5

2. What is the correct way to initialize a variable declared with the data type of Book, as a new Book object?

A. Book b;
B. Book b = new Book();
C. Book b = new Book[];
D. Book b = Book();

3. What is the difference between an int and an Integer?

A. Nothing. They are fully interchangeable.
B. An int is an object and Integer is a primitive. An int is fastest when performing calculations.
C. An int is a primitive and Integer is an object. An int is fastest when performing calculations.
D. This is a trick question. There is no such thing as an Integer.
E. This is a trick question. An Integer can be defined to be anything a developer wants it to be.

4. The number of arguments in a default constructor is:

a) 0
b) 1
c) The number of return types of the constructor
d) The number of instances that will be created

5. Memory is set aside for an object when the compiler encounters which keyword?
   a) continue
   b) create
   c) new
   d) break

**Answers:**

1. Answer:
D.Integer, Tree, and int[] are all references to objects. Remember an array is an object. A, B, C, E and F are incorrect.

2. Answer:
The correct way to declare an object is to use new and then the object name followed by parentheses. The parentheses are used to pass arguments to the constructor if needed. A, C, and D are incorrect. A is incorrect because it does not initialize a new Book object. C is incorrect because the square brackets are used instead of parentheses. D is incorrect because the new keyword is missing.

3. Answer:
C An int is a primitive, and primitives are faster when performing calculations. An Integer is an object. The capital letter 'I' should help you distinguish objects from primitives. A, B, D, and E are incorrect.

4. Answer: A

5. Answer: C

**Object Associations and Abstractions**

1. What associations are considered weak relationships? (Choose all that apply.)
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

2. What associations are considered strong relationships? (Choose all that apply.)
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

3. Which association can be said as object A has-an object B?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

4. Which association can be said as object A is part-of object B?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

5. Which association can be said as object A is composed-of object B?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

6. Which association has a life cycle responsibility for the object it contains?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

7. Association navigation is best described as which of the following?
   A. The ability to navigate, or access, an object that is contained in another object.
   B. The ability to search for and find an object that is contained in another object.
   C. The possibility of passing an object to another object via a method.
   D. The ability to invoke methods of an object that will then change the path of code execution.
   E. The ability to invoke methods of an object to determine the current path of execution.

8. What would the multiplicity be in the following relationship? A Lamp object has-a LightBulb object.
   A. One-to-one
   B. One-to-many
   C. Many-to-many

9. A composition association cannot exist in what multiplicity?
   A. One-to-one
   B. One-to-many
   C. Many-to-many

10. What would the multiplicity be in the following relationship? A BookShelf object has-a reference to an array made up of Book objects.
   A. One-to-one
   B. One-to-many
   C. Many-to-many

Use the following code example for the next four questions:

```
public class Client {
  Address address;
  AccountNum[] accountNums;
  void setAddress(Address newAddress) {
    address = newAddress;
  }
  public Client() {
    accountNums = new AccountNum[2];
    accountNums[0] = new AccountNum();
    accountNums[1] = new AccountNum();
  }
}
```

11. In the preceding code segment, what is the relationship of the Client object and the address variable?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

12. In the preceding code segment, what is the relationship of the Client object and the accountNums variable?
   A. Direct association
   B. Temporary association
   C. Composition association
   D. Aggregation association

13. In the preceding code segment, what is the multiplicity between the Client object and the address variable?

A. One-to-one
B. One-to-many
C. Many-to-many

14. In the preceding code segment, what is the multiplicity between the Client object and the accountNums variable?

A. One-to-one
B. One-to-many
C. Many-to-many

15. Which of the following statements are true? (Choose all that apply.)

A. Association navigation can be quad-directional.
B. Association navigation can be bidirectional.
C. Association navigation can have no direction.
D. Association navigation can be unidirectional.

16. What contains methods and instance variables and can be instantiated?

A. Concrete class
B. Abstract class
C. Java class
D. Interface

17. What is used to define a public interface?

A. Concrete class
B. Abstract class
C. Java class
D. Interface

18. What can contain unimplemented methods and instance variables and cannot be instantiated?

A. Concrete class
B. Abstract class
C. Java class

19. Inheritance provides which of the following? (Choose all that apply.)

A. Faster execution times since methods can inherit processor time from superclasses
B. Allows developers to place general code in a class that more specific classes can gain through inheritance
C. Promotes code reuse
D. Is an automated process to transfer old code to the latest Java version

20. What is a class being inherited referred to as? (Choose all that apply.)

A. Subclass
B. Superclass
C. Base class
D. Super duper class

21. Which of the following defines information hiding?

A. Information hiding is hiding as much detail about your class as possible so others can't steal it.
B. Information hiding is about hiding implementation details and protecting variables from being used the wrong way.
C. Information hiding is used to obscure the interworking of your class so external classes must use the public interface.

22. What is the proper signature for class X if it inherits class Z?

A. public class X inherits Z{ ... }
B. public class X extends Z{ ... }
C. public class X implements Z{ ... }

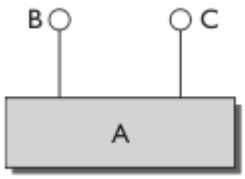
23. How many classes can a class extend directly?

A. Zero
B. One
C. Two
D. As many as it needs

24. How many interfaces can a class implement directly?

A. Zero
B. One
C. Two
D. As many as it needs

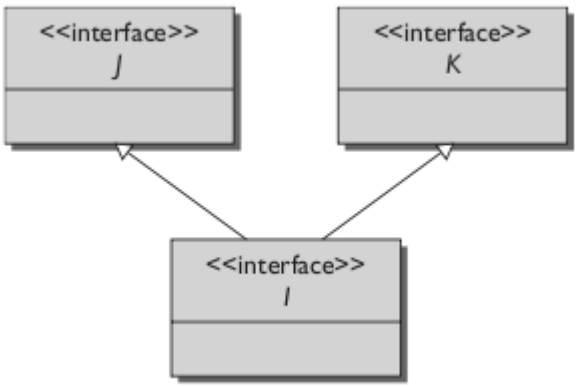
25. Consider the following UML illustration for assistance with this question:




What is the proper signature for class A if it implements interfaces B and C?

A. public class A implements B, implements C{ ... }
B. public class A implements B, C{ ... }
C. public class A interface B, interface C{ ... }
D. public class A interface B, C{ ... }
E. public class A extends B, C{ ... }

26. Consider the following UML illustration for assistance with this question:




What is the proper signature for interface I to inherit interfaces J and K?

A. public interface I extends J, K{ ... }
B. public interface I implements J, K{ ... }
C. public interface I implements J, implements K{ ... }
D. public interface I interface J, K{ ... }

**Answers:**

1. Answer:
A, B and D. Each of these associations is considered weak. This means that they do not have any life cycle responsibility and that if the relationship was lost, each object would still maintain its meaning. C is incorrect. This is an example of composition and has a strong relationship.

2. Answer:
C: This association is considered strong. It does have a life cycle responsibility and if the relationship was lost, each object would lose some or all of its meaning. A, B, and D are incorrect. They are examples of associations that have weak relationships.

3. Answer:
A. Direct association is a has-a relationship. B, C, and D are incorrect.

4. Answer:
D. Aggregation association is a part-of relationship. One object will be used to make up another object. However, neither object depends on the other for its existence and meaning. A, B, and C are incorrect.

5. Answer:
C. Composition association is a composed-of relationship. One object will be used to make up another object. If this relationship was lost, the meaning of the objects would also change. This is a strong relationship and has a life cycle responsibility for the inner object. A, B, and D are incorrect.

6. Answer:
C. Composition association has the responsibility to maintain the life cycle of the object that it is composed-of. A, B, and D are incorrect. They are all weak relationships that have no life cycle responsibility.

7. Answer:
A. Association navigation is the ability to access an object that is contained in another.

8. Answer:
A. There is one Lamp object and one LightBulb object, therefore this is one-to-one.

9. Answer:
C. is the correct answer. Composition association requires that it have responsibility for the life cycle of the objects it is composed-of. It is impossible to have this responsibility in a many-to-many relationship because many objects would contain references to all of the objects. A and B are incorrect.

10. Answer:
B. BookShelf is a single object and contains a reference to an array of Book objects. The key to this question is the fact that you are dealing with an array. This means there are many Book objects; therefore it is a one-to-many relationship. A. and C are incorrect.

11. Answer:
A. Direct association is the best answer, because logically a Client object has-an Address object. B, C, and D are incorrect.

12. Answer:
C. This is a composition association because the Client object is composed-of the AccountNum objects. This is a strong relationship since the Client object maintains the life cycle of the AccountNum objects. A, B, and D are incorrect.

13. Answer:
A. is the correct answer. Since there are no arrays or collections involved with either the Client object or Address object, this must be one-to-one. B and C are incorrect.

14. Answer:
B. The array of AccountNum objects should be a giveaway that this is one-to-many. A and C are incorrect.

15. Answer:
B and D. A relationship can only be unidirectional or bidirectional. A and C are incorrect.

16. Answer:
A. A concrete class is the standard Java class that is used to create objects. B, C, and D are incorrect. B is incorrect because an abstract class cannot be instantiated. C is incorrect because a Java class is a made-up term. D is incorrect because an interface does not contain methods and cannot be instantiated.

17 Answer:
D. An interface is used to define a public list of methods that must be implemented by the class. This represents a public interface. A, B, and C are incorrect. A is incorrect because a concrete class is used to build objects. B is incorrect because abstract classes are used to define abstract methods for other classes to override. C is incorrect because a Java class is a made-up term.

18 Answer:
B. An abstract class must always be extended; it cannot be instantiated to create an object. It can contain implemented and unimplemented methods. A and C are incorrect. A is incorrect because a concrete class is not able to have any unimplemented methods. C is incorrect because a Java class is a made-up term.

19 Answer:
B. and C. Both statements are true about inheritance. A and D are incorrect. A is incorrect because inheritance has no effect on processor scheduling. D is incorrect because inheritance has no relationship to the Java version.

20 Answer:
B. and C. The class that is inherited is the base class or superclass in reference to the class that extends it. A and D are incorrect. A is incorrect because the subclass is the class that inherits from another. D is incorrect because this is a made-up term.

21 Answer:
B. Good class design hides as many methods and instance variables as possible. This is done by using the private access modifier. This is so external objects do not try to interact with the object in ways the developer has not intended. Hiding information makes code easier to maintain and more modular. A and C are incorrect. A is incorrect because information hiding has nothing to do with protecting your code from others. C is incorrect because access modifiers should be used to force external classes to use the proper public interface.

22. Answer:
B . The extends keyword is used to inherit a class. A and C are incorrect. A is incorrect because inherits is not a valid Java keyword. C is incorrect because the implements keyword is used for interfaces, not classes.

23 Answer:
B. A class can only extend one other class. However, it is possible to have one class extend a class that extends another class, and so on. A, C, and D are incorrect.

24 Answer:
D. Unlike extending other classes, a class can implement as many interfaces as it needs. A, B, and C are incorrect.

25 Answer:
B.A class uses the keyword implements to implement an interface. To implement multiple interfaces, they are shown in a comma-delimited list after the keyword implements. A, C, D, and E are incorrect. A is incorrect because the implements keyword should not be listed more than once. C is incorrect because the implements keyword should be used instead of interface, and it should be listed only once. D is incorrect because implements should be used instead of interface. E is incorrect because extends is used for classes not interfaces; implements should be used instead.

26 Answer:
A.An interface can also inherit other interfaces. Unlike classes, they can inherit or extend as many other interfaces as needed. An interface uses the keyword extends, followed by a comma-delimited list of all the other interfaces it wants to extend. B, C, and D are incorrect. B is incorrect because only classes implement interfaces. An interface extends other interfaces. C is incorrect because extends should be used and only listed once. D is incorrect because the interface keyword is not used correctly.

**Conditional Logic and Iterative Loops**

1. Given:

```
public class ArithmeticResultsOutput {
  public static void main (String[] args) {
    int i = 0;
    int j = 0;
    if (i++ == ++j) {
      System.out.println("True: i=" + i + ", j=" + j);
    } else {
      System.out.println("False: i=" + i + ", j=" + j);
    }
  }
}
```

What will be printed out?

A. True: i=0, j=1
B. True: i=1, j=1
C. False: i=0, j=1
D. False: i=1, j=1

2. Which set of operators represents the complete set of valid Java assignment operators?

A. %=, &=, *=, $=, :=, /=, ^=, |=, +=, <<=, =, -=, >>=, >>>=
B. %=, &=, *=, /=, ^=, |=, +=, <<=, <<<=, =, -=, >>=, >>>=
C. %=, &=, *=, /=, ^=, |=, +=, <<=, =, -=, >>=, >>>=
D. %=, &=, *=, $=, /=, ^=, |=, +=, <<=, <<<=, =, -=, >>=, >>>=

3. Given the following Java code segment, what will be printed, considering the usage of the modulus operators?

```
System.out.print(49 % 26 % 5 % 1);
```

A. 23
B. 3
C. 1
D. 0

4. Given:

```
public class BooleanResultsOutput {
  public static void main (String[] args) {
    boolean booleanValue1 = true;
    boolean booleanValue2 = false;
    System.out.print(!(booleanValue1 & !booleanValue2) + ", ");
    System.out.print(!(booleanValue1 | !booleanValue2)+ ", ");
    System.out.print(!(booleanValue1 ^ !booleanValue2));
  }
}
```

What will be printed, considering the usage of the logical Boolean operators?

A. false, false, true
B. false, true, true
C. true, false, true
D. true, true, true

5. Given:

```
public class ArithmeticResultsOutput {
  public static void main (String[] args)  {
    int i1 = 100; int j1 = 200;
    if ((i1 == 99) & (--j1 == 199)) {
      System.out.print("Value1: " + (i1 + j1) + " ");
    } else {
      System.out.print("Value2: " + (i1 + j1) + " ");
    }
    int i2 = 100; int j2 = 200;
    if ((i2 == 99) && (--j2 == 199)) {
      System.out.print("Value1: " + (i2 + j2) + " ");
    } else {
      System.out.print("Value2: " + (i2 + j2) + " ");
    }
    int i3 = 100; int j3 = 200;
    if ((i3 == 100) | (--j3 == 200)) {
      System.out.print("Value1: " + (i3 + j3) + " ");
    } else {
      System.out.print("Value2: " + (i3 + j3) + " ");
    }
    int i4 = 100; int j4 = 200;
    if ((i4 == 100) || (--j4 == 200)) {
      System.out.print("Value1: " + (i4 + j4) + " ");
     } else {
      System.out.print("Value2: " + (i4 + j4) + " ");
    }
  }
}
```

What will be printed out?

A. Value2: 300 Value2: 300 Value1: 300 Value1: 300
B. Value2: 299 Value2: 300 Value1: 299 Value1: 300
C. Value1: 299 Value1: 300 Value2: 299 Value2: 300
D. Value1: 300 Value1: 299 Value2: 300 Value2: 299

6. Given the following code segment:

```
public void validatePrime() {
  long p = 17496; // 'prime number' candidate
  Double primeSquareRoot = Math.sqrt(p);
  boolean isPrime = true;
  for (long j = 2; j <= primeSquareRoot.longValue(); j++) {
    if (p % j == 0) {
      // Print divisors
      System.out.println(j + "x" + p / j);
      isPrime = false;
    }
  }
  System.out.println("Prime number: " + isPrime);
}
```

Which of the following is true? Hint: 17496 is not a prime number.

A. The code will not compile due to a syntactical error somewhere in the code.
B. The code will not compile since the expression (p % j == 0) should be written as ((p % j) == 0).
C. Divisors will be printed out (for example,2x8478, and so on), along with “Prime number: false” as the final output.
D. Divisors will be printed out (for example,2x8478, and so on), along with “Prime number: 0” as the final output.

7. Given:

```
public class EqualityTests {
  public static void main (String[] args) {
    Integer value1 = new Integer("312");
    Integer value2 = new Integer("312");
    Object object1 = new Object();
    Object object2 = new Object();
    Object object3 = value1;
  }
}
```

Which expressions evaluate to true?

A. value1.equals(value2)
B. value1.equals(object1)
C. value1.equals(object3)
D. object1.equals(object2)

8. Given x is declared with a valid integer, which conditional statement will not compile?

A. if (x == 0) {System.out.println("True Statement");}
B. if (x == 0) {System.out.println("False Statement");}
C. if (x == 0) {;} elseif (x == 1) {System.out.println("Valid Statement");}
D. if (x == 0) ; else if (x == 1){} else {;}

9. Which is not a type of statement?

A. Conditional statement
B. Assignment statement
C. Iteration statement
D. Propagation statement

10. You need to update a value of a hash table (that is, HashMap) where the primary key must equal a specified string. Which statements would you need to use in the implementation of this algorithm?

A. Iteration statement
B. Expression statement
C. Conditional statement
D. Transfer of control statement

11. Which keyword is part of a transfer of control statement?

A. if
B. return
C. do
D. assert

12. A switch statement works with which wrapper class/reference type?

A. Character
B. Byte
C. Short
D. Int

13. Which statements correctly declare boolean variables?

A. Boolean isValid = true;
B. boolean isValid = TRUE;
C. boolean isValid = new Boolean (true);
D. boolean isValid = 1;

14. Which of the following statements will not compile?

A. if (true) ;
B. if (true) {}
C. if (true) {:}
D. if (true) {;}

15. The for loop has been enhanced in Java 5.0. Which is NOT a common term for the improved for loop.

A. The “for in” loop
B. The specialized for loop
C. The “for each” loop
D. The enhanced for loop

16. Given:

```
COUNTER := 1
WHILE COUNTER LESS THAN 10
 PRINT COUNTER AND A NEW LINE
 COUNTER := COUNTER + 1
ENDWHILE
```

Which Java code segment implements the pseudo-code algorithm?

A.
```
INT counter = 1;
WHILE (counter < 10) {
   System.out.print(counter + "\n");
   counter++;
}
```

B.
```
int counter = 1;
while {counter < 10} {
   System.out.print(counter + "\n");
   counter++;
}
```

C.
```
int counter = 1;
while (counter < 10) {
   System.out.println(counter);
   counter++;
}
```

D.
```
int counter = 1;
while (counter < 10) {
   System.out.println(counter + "\n");
   counter++;
}
```

**Answers:**

1. Answer:

D. The value of j is prefix incremented before the evaluation; however, the value of i is not. Therefore, the expression is evaluated with a boolean value of false as a result since 0 does not equal 1 (that is, i=0 and j=1). After the expression has been evaluated, but before the associated print statement is executed, the value of i is postfix incremented (that is, (i=1)). Therefore, the correct answer is False: i=1, j=1). A, B, and C are incorrect answers as justified by the correct answer's explanation.

2. Answer:
   C. The complete set of valid Java assignment operators is represented. A, B, and D are incorrect answers. A is incorrect since $= and := are not valid Java assignment operators. B is incorrect because <<<= is not a valid Java assignment operator. D is incorrect because $= and <<<= are not valid Java assignment operators.

3. Answer:
D. The remainder of 49/26 is 23. The remainder of 23/5 is 3. The remainder of 3/1 is 0. The answer is 0.
A , B, and C are incorrect answers as justified by the correct answer's explanation.

4. Answer:
A. The first expression statement (!(true & !(false))) evaluates to false. Here, the right operand is negated to true by the (Boolean invert) operator, the Boolean AND operator equates the expression of the two operands to true, and the (Boolean invert) operator equates the resultant value to false. The second expression statement (!(true | !(false))) evaluates to false. Here, the right operand is negated to true by the (Boolean invert) operator, the Boolean OR operator equates the expression of the two operands to true, and the (Boolean invert) operator equates the resultant value to false. The third expression statement (!(true ^ !(false))) evaluates to true. Here, the right operand is negated to true by the (Boolean invert) operator, the Boolean XOR operator equates the expression of the two operands to false, and the (Boolean invert) operator equates the resultant value to true. B, C, and D are incorrect answers as justified by the correct answer's explanation.

5. Answer:
B. is the correct because Value2: 299 Value2: 300 Value1: 299 Value1: 300 will be printed to the standard out. Note that && and || are short-circuit operators. So... When the first operand of a conditional AND (&&) expression evaluates to false, the second operand is not evaluated. When the first operand of a conditional OR (||) expression evaluates to true, the second operand is not evaluated. Thus, for the second and fourth if statements, the second operand isn't evaluated. Therefore, the prefix increment operators are never executed and do not affect the values of the j[x] variables. A, C, and D are incorrect answers as justified by the correct answer's explanation.

6. Answer:
C. Divisors will be printed out followed by "Prime number: false". For those curious, the complete list of divisors printed are 2x8748, 3x5832, 4x4374, 6x2916, 8x2187, 9x1944, 12x1458, 18x972, 24x729, 27x648, 36x486, 54x324, 72x243, 81x216, and 108x162. A , B, and D are incorrect answers. A is incorrect because there are no syntactical errors A in the code. B is incorrect because a set of parentheses around "p % j" is not required. Answer D is incorrect because the code does not print out the character 0, it prints out the boolean literal value false.

7. Answer:
A. and C. A is correct because the class Integer implements the Comparable interface, allowing use of the equals method. C is correct because the Integer object was used to create the Object reference. B and D are incorrect because the code cannot equate two objects with different references.

8.Answer:
C. The statement will not compile. Without a space between the else and if keywords, the compiler will be thrown an error similar to "Error: method elseif (boolean) not found..." A, B, and D are incorrect. All of these conditional statements will compile successfully.

9. Answer:
D. There is no such thing as a propagation statement. A, B, and C are incorrect. Conditional, assignment, and iteration are all types of statements.

10. Answer:
A, B, and C. An Iteration, expression, and conditional statements would be used to implement the algorithm. The following code segment demonstrates the use of these statements by programmatically replacing the ring on the little finger of a person's left hand. The statements are prefaced by comments that identify their types.

```
import java.util.HashMap;
public class HashMapExample {
   public static void main(String[] args) {
     HashMap<String,String> leftHand = new HashMap<String,String>();
     leftHand.put("Thumb", null);
     leftHand.put("Index finger", "Puzzle Ring");
     leftHand.put("Middle finger", null);
     leftHand.put("Ring finger", "Engagement Ring");
     leftHand.put("Little finger", "Pinky Ring");
     // Iteration statement
     for (String s : leftHand.keySet()) {
       // Conditional statement
       if (s.equals("Little finger")) {
          System.out.println(s + " had a " + leftHand.get(s));
          // Expression Statement
          leftHand.put("Little finger", "Engineer's Ring");
         System.out.println(s + " has an " + leftHand.get(s));
       }
     }
   }
}
$ Little finger had a Pinky Ring
$ Little finger has an Engineer's Ring
```

D is incorrect. There is no transfer of control statement in the algorithm.

11. Answer:
B. The keyword return is used as part of a transfer of control statement. A, C, and D are incorrect. The keywords if, do, and assert are not part of any transfer of control statements.

12. Answer:
A, B, and C. The switch statements work with Character, Byte, and Short wrapper classes as well as the Integer wrapper class. D is incorrect. There is no such thing as an Int wrapper type. This was a trick question. The switch statement works with either the int primitive or the Integer wrapper type.

13. Answer:
A. and C. These statements properly declare boolean variables. Remember, the only valid literal values for the boolean primitives are true and false. B and D are incorrect. B is incorrect because TRUE is not a valid literal value. D is incorrect because you cannot assign the value 1 to a boolean variable.

14. Answer:
C. A colon is invalid by itself. A, B, and D are incorrect. All of the statements represent compilable code.

15. Answer:
B. The enhanced for loop is not commonly referenced as a specialized for loop. A, C, and D are incorrect. The enhanced for loop is also commonly referenced as the for in loop and the for each loop.

16. Answer:
C.The answer implements the pseudo-code algorithm correctly. A, B, and D are incorrect. Answer A is incorrect because INT and WHILE are not Java keywords. B is incorrect because the expression for the while statement is enclosed in braces where parentheses are expected. D is incorrect because two new lines are printed, one with (\n) and one with the println method.

**Coding Questions**

Code Question1 (Fundamental Data Types):

```
1. public class InitialTest {
2. int x;
3. public static void main(String[] args) {
4. new InitialTest().printIt();
5. }
6. public void printIt(){
7. int y;
8. int z;
9. y=2;
10. System.out.println(x +" "+ y);
11. // System.out.println(z);
12. }
13. }
```

Step 1 Run and compile (note – you will need to call your java file "InitialTest.java", also do not forget to remove the numbers 1. - 13.)

Step 2 Uncomment line 11 and try to compile. The code will generate a compiler error because you are using the local variable z without initializing it. Note that lines 7 and 8 did not generate compiler errors. That means you do not have to initialize a local variable in the same statement where you declare it. However, you must initialize it before using it.

Step 3 Initialise the variable value z so that the code will compile and run

Step 4 add a char variable in the printIt() method and initialise it with the value 'c' and print out its contents to the output screen.

Step 4 add a byte variable in the printIt() method and initialise it with the value 180 and print out its contents to the output screen and try to compile. What was the result? Why?

Step 5 now initialise the byte to 120 and compile

**Coding Question 2 (Conditional Logic)**

```
public class ArithmeticResultsOutput {
      public static void main (String[] args) {
        int i = 0;
        int j = 0;

          if (i>j){
          System.out.println("i is bigger than  j");
          }
          else
          {
          System.out.println("i is not bigger than  j");
          }

        if (i++ == ++j) {
          System.out.println("True: i=" + i + ", j=" + j);
        } else {
          System.out.println("False: i=" + i + ", j=" + j);
        }
      }
    }
```

Step 1 run and compile the code.

Step 2 – try to step through the code and understand the output. Feel free to modify it to give different results.

**Coding Question 3 (Loops)**

Write a class and main method that uses (a) a for loop (b) a while loop and (c) a do while loop to print out the values 1 to 10.