Appendix A. List of Practice Problems and Feedback

Short Description

Arithmetic Plus Precedes 1 (CC Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. What is the value printed to the console?

```
public class Example{
   public static void main(String[] args){
      String a = 1 + 2 + "AAA" + 3 + 4;
      System.out.println(a);
   }
}
```

Hint

Concatenation has the same order of precedence than addition of integers.

Answer 1 (Correct)

3AAA34

The values are concatenated to a larger sequence of characters from left to right.

Feedback 1

You chose the correct option

Answer 2

3AAA34

The integers are first added, then concatenated to a larger sequence of characters.

Feedback 2

You chose the incorrect option

Answer 3

3AAA7

The values are concatenated to a larger sequence of characters from left to right.

Feedback 3

You chose the incorrect option

Answer 4

3AAA7

The integers are first added, then concatenated to a larger sequence of characters.

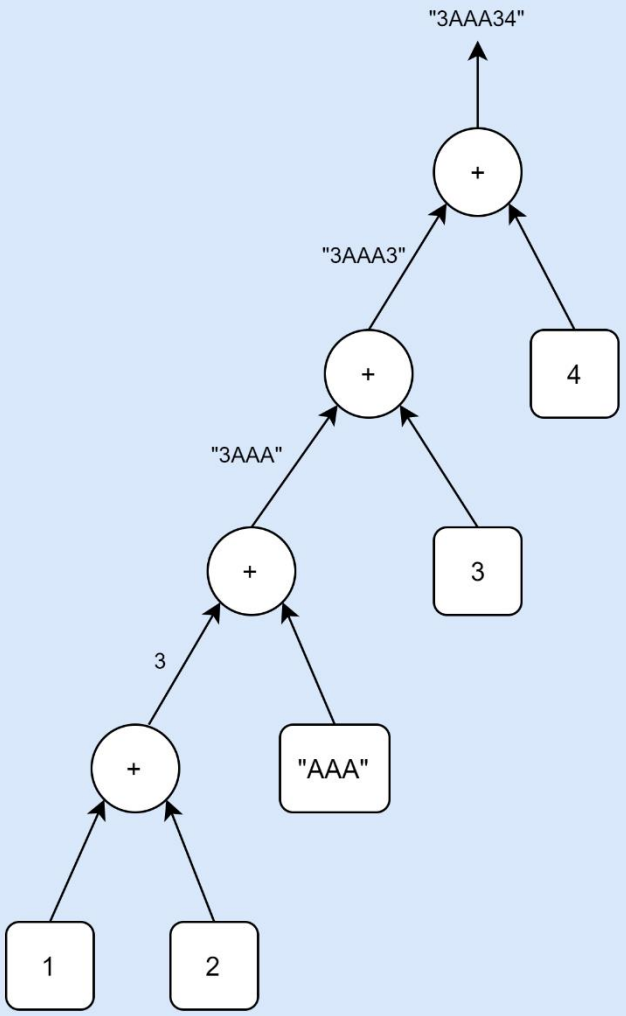
Feedback 4

You chose the incorrect option

Overall Feedback

Valid Conception

On line 3, several things are added together to create String a. Both addition of integers and concatenation of strings have the same order of precedence, therefore the resulting string printed is "3AAA34".



Short Description

Arithmetic Plus Precedes 1 (LT Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. A student claims that the value printed to the console is "3AAA34". Is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      String a = 1 + 2 + "AAA" + 3 + 4;
      System.out.println(a);
   }
}
```

Hint

Concatenation has the same order of precedence than addition of integers.

Answer 1 (Correct)

3AAA34

The values are concatenated to a larger sequence of characters from left to right.

Feedback 1

The student is correct.

Answer 2

3AAA34

The integers are first added, then concatenated to a larger sequence of characters.

Feedback 2

The student may still be confused.

Answer 3

3AAA7

The values are concatenated to a larger sequence of characters from left to right.

Feedback 3

The student may still be confused.

Answer 4

3AAA7

The integers are first added, then concatenated to a larger sequence of characters.

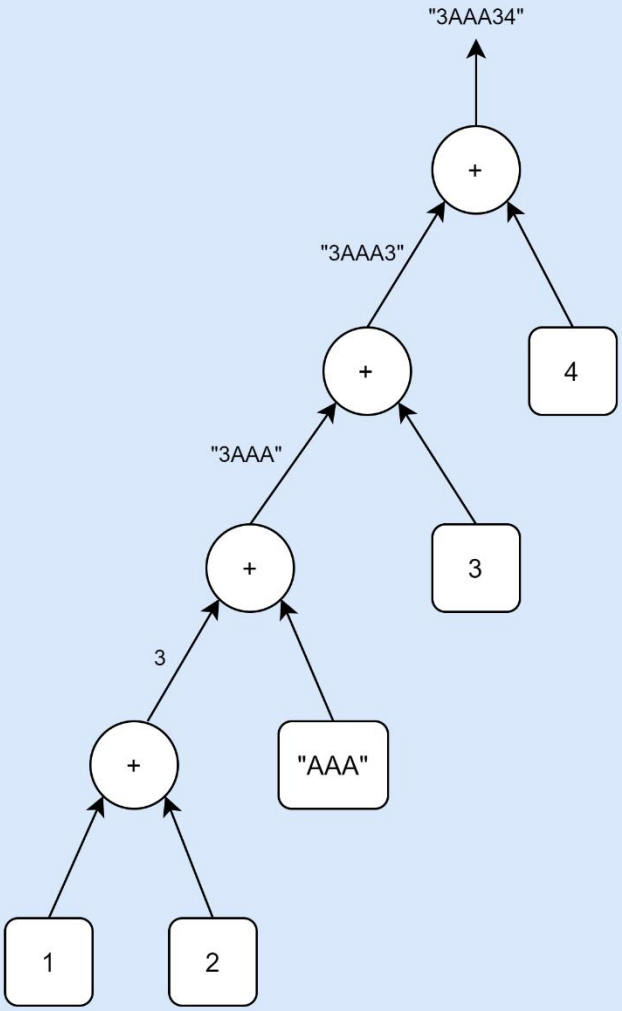
Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

On line 3, several things are added together to create String a. Both addition of integers and concatenation of strings have the same order of precedence, therefore the resulting string printed is "3AAA34" and the student is correct.



Short Description

Arithmetic Plus Precedes 2 (CC Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. What is the value printed to the console?

```
public class Example{
   public static void main(String[] args){
      String a = 1 + 1 + "ABC" + 3 * 2 + 5 + "DEF";
      System.out.println(a);
   }
}
```

Hint

Multiplication has a higher order of precedence than concatenation or addition of integers.

Answer 1 (Correct)

2ABC65DEF

The integers are multiplied first, then addition of integers and concatenation of characters is performed second from left to right.

Feedback 1

You chose the correct option

Answer 2

Compilation error

The multiplication cannot be performed on a string literal and integer as operands, the use of parentheses is necessary to resolve the error.

Feedback 2

You chose the incorrect option

Answer 3

2ABC11DEF

The addition and multiplication operations on integers are performed first, then concatenated with the string literals second.

Feedback 3

You chose the incorrect option

Answer 4

2ABC21DEF

The addition of integers is performed first, followed by the multiplication, and then concatenation of string literals.

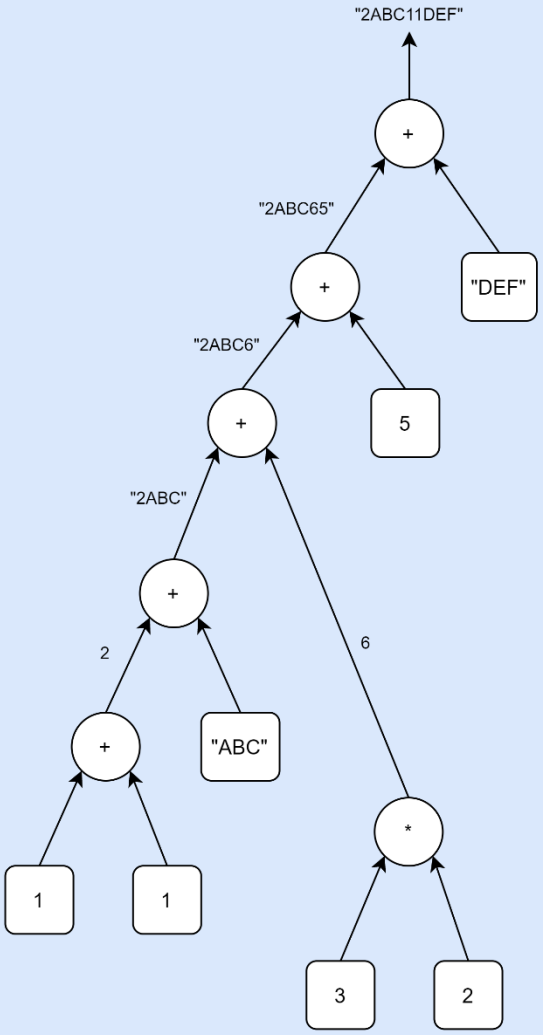
Feedback 4

You chose the incorrect option

Overall Feedback

Valid Conception

On line 3, several things are added together to create String a. The order of precedence is as follows: multiplication, and then concatenation and addition have the same order. Therefore the resulting string is "2ABC65DEF"



Short Description

Arithmetic Plus Precedes 2 (LT Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. A student claims that the value is "2ABC11DEF", is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      String a = 1 + 1 + "ABC" + 3 * 2 + 5 + "DEF";
      System.out.println(a);
   }
}
```

Hint

Multiplication has a higher order of precedence than concatenation or addition of integers.

Answer 1 (Correct)

2ABC65DEF

The integers are multiplied first, then addition of integers and concatenation of characters is performed second from left to right.

Feedback 1

The student is therefore incorrect

Answer 2

Compilation error

The multiplication cannot be performed on a string literal and integer as operands, the use of parentheses is necessary to resolve the error.

Feedback 2

The student may still be confused.

Answer 3

2ABC11DEF

The addition and multiplication operations on integers are performed first, then concatenated with the string literals second.

Feedback 3

The student may still be confused.

Answer 4

2ABC21DEF

The addition of integers is performed first, followed by the multiplication, and then concatenation of string literals.

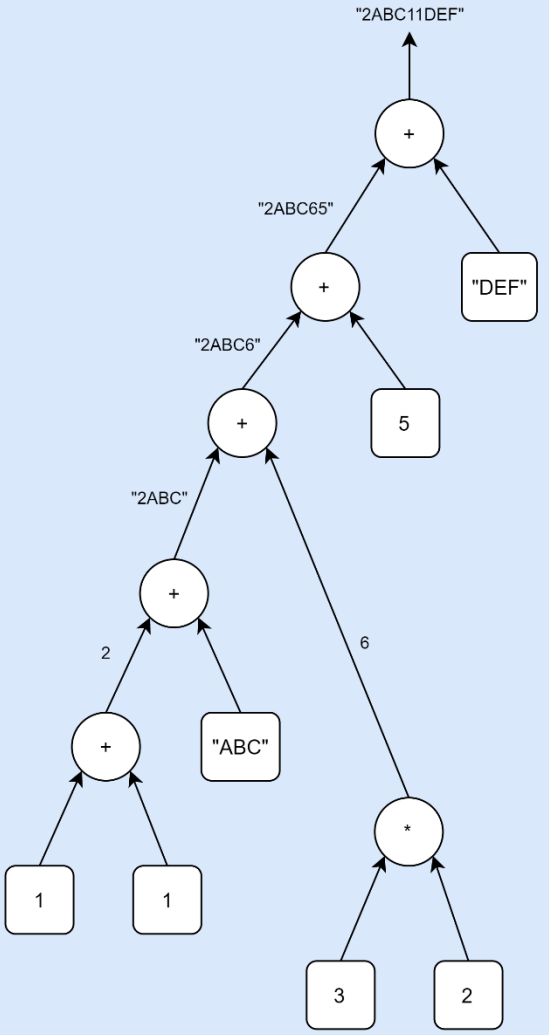
Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

On line 3, several things are added together to create String a. The order of precedence is as follows: multiplication, and then concatenation and addition have the same order. Therefore the resulting string is "2ABC65DEF" and the student is incorrect



Short Description

Arithmetic Plus Precedes 3 (CC Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. What is the value printed to the console?

```
public class Example{
   public static void main(String[] args){
      String a = "50" + 50 + 100;
      System.out.println(a);
   }
}
```

Hint

The addition of integers has the same order of precedence than concatenation of String literals.

Answer 1 (Correct)

5050100

The concatenation of string literals occurs first, from left to right through each operand.

Feedback 1

You chose the correct option

Answer 2

50150

The addition of integers is performed first, then the concatenation of string literals.

Feedback 2

You chose the incorrect option

Answer 3

5050100

The addition of integers is performed first, then the concatenation of string literals.

Feedback 3

You chose the incorrect option

Answer 4

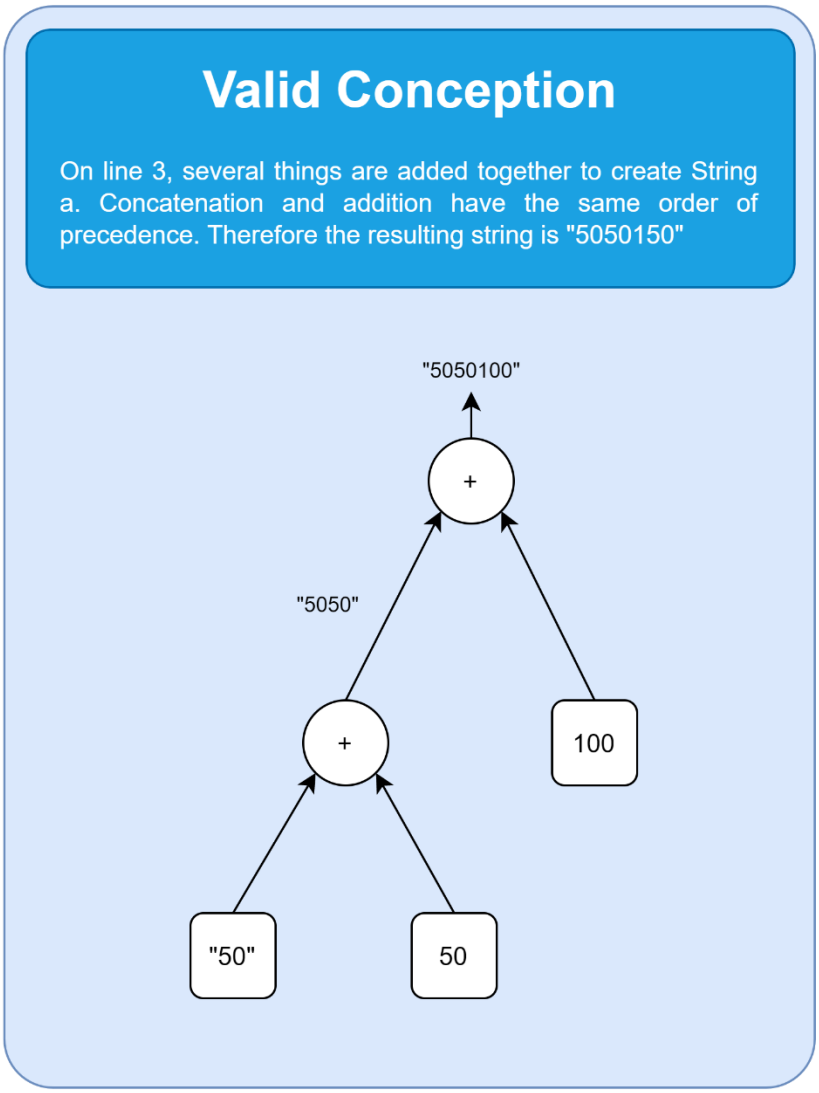
50150

The concatenation of string literals occurs first, from left to right through each operand.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

Arithmetic Plus Precedes 3 (LT Lecture 3 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. A student claims that the value is "5050100", is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      String a = "50" + 50 + 100;
      System.out.println(a);
   }
}
```

Hint

The addition of integers has the same order of precedence than concatenation of String literals.

Answer 1 (Correct)

5050100

The concatenation of string literals occurs first, from left to right through each operand.

Feedback 1

The student is therefore correct

Answer 2

50150

The addition of integers is performed first, then the concatenation of string literals.

Feedback 2

The student may still be confused.

Answer 3

5050100

The addition of integers is performed first, then the concatenation of string literals.

Feedback 3

The student may still be confused.

Answer 4

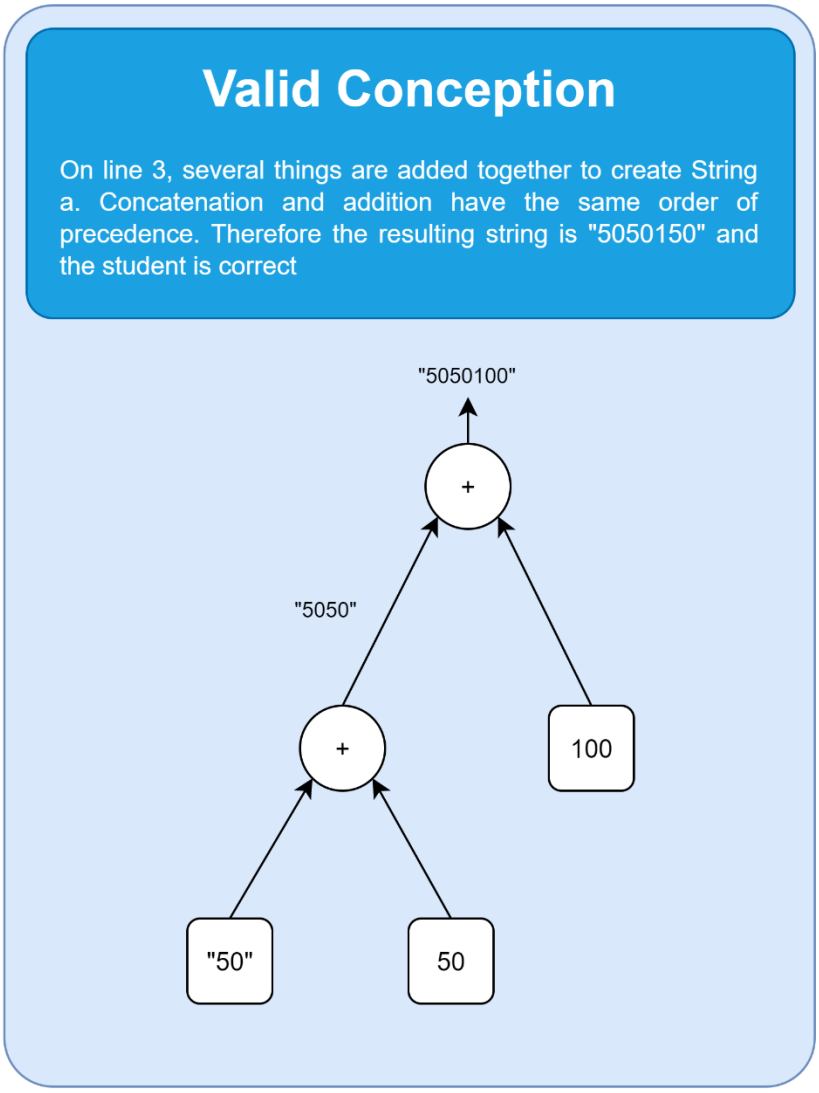
50150

The concatenation of string literals occurs first, from left to right through each operand.

Feedback 4

The student may still be confused.

Overall Feedback



Short Description

Postfix Increment (CC Lecture 4 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. What are the values printed to the console?

```
public class Example {
   public static void main(String[] args) {
      int a = 1;
      int b = a++;
      System.out.println(a);
      System.out.println(b);
   }
}
```

Hint

The post fix increment operation returns the original value prior to the arithmetic operation.

Answer 1 (Correct)

2

1

The original value of a is assigned to b prior to incrementing the value of a. Then, the value of b is set to a added to one, resulting in 2.

Feedback 1

You chose the correct option

Answer 2

1

2

The value of a is assigned to 1. Then, the value of b is set to the addition of a and 1, resulting in 2.

Feedback 2

You chose the incorrect option

Answer 3

2

2

The updated value of a is assigned to b after incrementing the value of a. The value of b is set to the addition of a and 1, resulting in 2.

Feedback 3

You chose the incorrect option

Answer 4

1

2

The updated value of a is assigned to b after incrementing the value of a. The value of b is set to the addition of a and 1, resulting in 2.

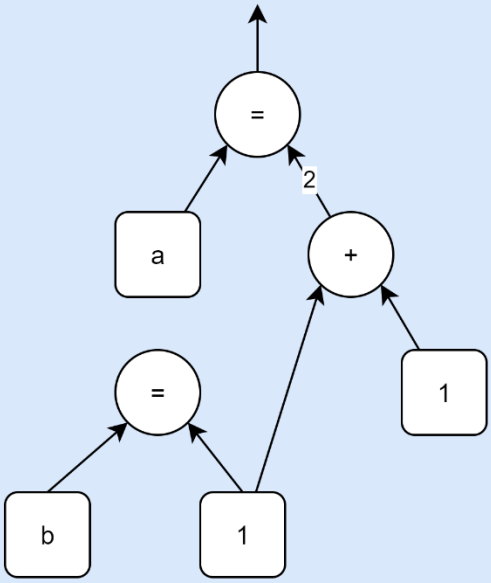
Feedback 4

You chose the incorrect option

Overall Feedback

Valid Conception

The original value of a is assigned to b prior to incrementing the value of a. Then, the value of a is set to a added to one, resulting in 2. The values printed to the console are 2 and 1.



Short Description

Postfix Increment (LT Lecture 4 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArithmeticPlusPrecedes/

Question Text

Trace the execution of this program. A student claims that the values printed to the console are 1 and 2. Is this correct? Justify your answer.

```
public class Example {
   public static void main(String[] args) {
      int a = 1;
      int b = a++;
      System.out.println(a);
      System.out.println(b);
   }
}
```

Hint

The post fix increment operation returns the original value prior to the arithmetic operation.

Answer 1 (Correct)

2

1

The original value of a is assigned to b prior to incrementing the value of a. Then, the value of b is set to a added to one, resulting in 2.

Feedback 1

The student is therefore incorrect.

Answer 2

1

2

The value of a is assigned to 1. Then, the value of b is set to the addition of a and 1, resulting in 2.

Feedback 2

The student may still be confused.

Answer 3

2

2

The updated value of a is assigned to b after incrementing the value of a. The value of b is set to the addition of a and 1, resulting in 2.

Feedback 3

The student may still be confused.

Answer 4

1

2

The updated value of a is assigned to b after incrementing the value of a. The value of b is set to the addition of a and 1, resulting in 2.

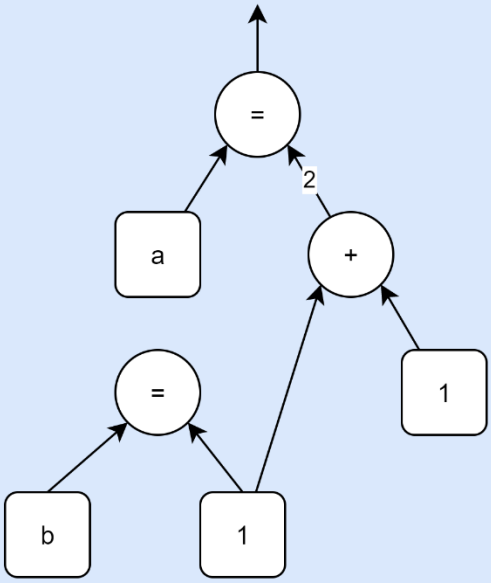
Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

The original value of a is assigned to b prior to incrementing the value of a. Then, the value of a is set to a added to one, resulting in 2. The values printed to the console are 2 and 1.



Short Description

Char Not Numeric (CC Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/CharNotNumeric/

Question Text

Trace the execution of this program. What is the value printed to the console?

```
public class CharNotNumeric {
   public static void main(String[] args){
      char a = 'A';
      char b = 'B';
      char c = 'C';
      if(c > b || b > a){
         System.out.println("The letter appears after in the alphabet.");
      }
   }
}
```

Hint

The character to decimal conversion is done implicitly.

Answer 1 (Correct)

The letter appears after in the alphabet

The conversion from character to decimal representation is implicit, therefore, the first operand of the OR condition returns true.

Feedback 1

You chose the correct option

Answer 2

The letter appears after in the alphabet

The conversion from character to decimal representation is implicit, therefore, both operand of the OR condition returns true.

Feedback 2

You chose the incorrect option

Answer 3

No value

The conversion from character to decimal representation is implicit, therefore, both operand of the OR condition returns false.

Feedback 3

You chose the incorrect option

Answer 4

Compilation error

The relational operators cannot be applied to characters, use .equals( ) method instead.

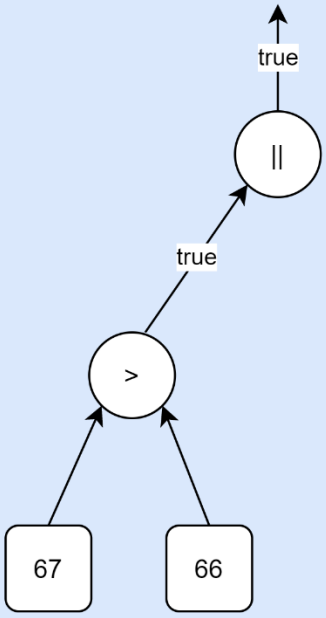
Feedback 4

You chose the incorrect option

Overall Feedback

Valid Conception

The conversion from character to decimal representation is implicit, therefore, the first operand of the OR condition returns true.



Short Description

Char Not Numeric (LT Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/CharNotNumeric/

Question Text

Trace the execution of this program. A student claims that the value printed to the console is "The letter appears after in the alphabet", is this correct? Justify your answer.

```
public class CharNotNumeric {
   public static void main(String[] args){
      char a = 'A';
      char b = 'B';
      char c = 'C';
      if(c > b || b > a){
         System.out.println("The letter appears after in the alphabet.");
      }
   }
}
```

Hint

The character to decimal conversion is done implicitly.

Answer 1 (Correct)

The letter appears after in the alphabet

The conversion from character to decimal representation is implicit, therefore, the first operand of the OR condition returns true.

Feedback 1

The student is therefore correct

Answer 2

The letter appears after in the alphabet

The conversion from character to decimal representation is implicit, therefore, both operand of the OR condition returns true.

Feedback 2

The student may still be confused

Answer 3

No value

The conversion from character to decimal representation is implicit, therefore, both operand of the OR condition returns false.

Feedback 3

The student may still be confused

Answer 4

Compilation error

The relational operators cannot be applied to characters, use .equals( ) method instead.

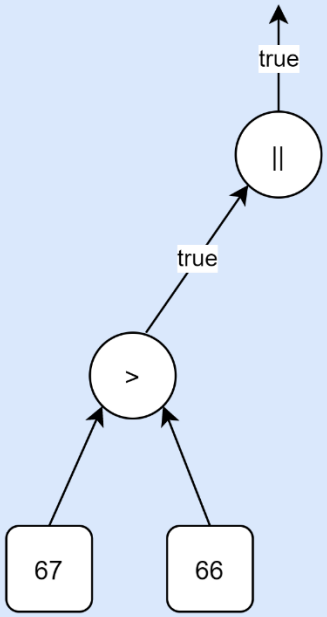
Feedback 4

The student may still be confused

Overall Feedback

Valid Conception

The conversion from character to decimal representation is implicit, therefore, the first operand of the OR condition returns true. The student is correct.



Short Description

Reference Variable Comparison - String (CC Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/EqualsComparesReferences/

Question Text

Trace the execution of this program. What values are printed to the console?

```
public class Example{
   public static void main(String[] args){
      String a = "String";
      String b = "String";
      System.out.println(a == b);
      System.out.println(a.equals(b));
   }
}
```

Hint

Reference variables hold the address for the String literal stored in heap memory.

Answer 1 (Correct)

true

true

The String literal pool optimizes memory by allocating the same address to string literals with the same sequence of characters. Therefore, both the comparison of the address and characters return true.

Feedback 1

You chose the correct option

Answer 2

true

true

The comparison of String literal values return true in both cases since the sequence of characters is identical.

Feedback 2

You chose the incorrect option

Answer 3

false

true

String literals are assigned different addresses in heap memory. Therefore, the comparison of their addresses return false, but the characters return true.

Feedback 3

You chose the incorrect option

Answer 4

false

true

You cannot use the equality relational operator to compare String literals, instead rely on the .equals( ) method.

Feedback 4

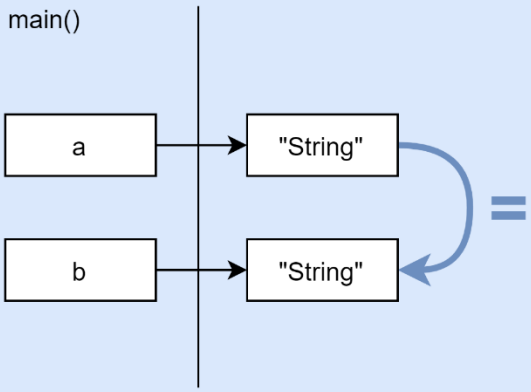
You chose the incorrect option

Overall Feedback

Valid Conception

Unlike with other objects, in the case with Strings both == and the .equals() method check the equality of the value contained by each variable. Therefore, both statements check the same thing so the following truth values are returned:

true
true



Short Description

Reference Variable Comparison - String (LT Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/EqualsComparesReferences/

Question Text

Trace the execution of this program. A student claims that the Boolean values printed to the console are true and true. Explain whether or not the student is correct, and justify your answer.

```
public class Example{
   public static void main(String[] args){
      String a = "String";
      String b = "String";
      System.out.println(a == b);
      System.out.println(a.equals(b));
   }
}
```

Hint

Reference variables hold the address for the String literal stored in heap memory.

Answer 1 (Correct)

true

true

The String literal pool optimizes memory by allocating the same address to string literals with the same sequence of characters. Therefore, both the comparison of the address and characters return true.

Feedback 1

The student is therefore correct.

Answer 2

true

true

The comparison of String literal values return true in both cases since the sequence of characters is identical.

Feedback 2

The student may still be confused.

Answer 3

false

true

String literals are assigned different addresses in heap memory. Therefore, the comparison of their addresses return false, but the characters return true.

Feedback 3

The student may still be confused.

Answer 4

false

true

You cannot use the equality relational operator to compare String literals, instead rely on the .equals( ) method.

Feedback 4

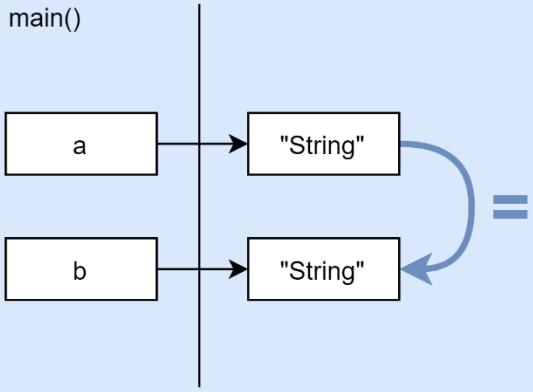
The student may still be confused.

Overall Feedback

Valid Conception

Unlike with other objects, in the case with Strings both == and the .equals() method check the equality of the value contained by each variable. Therefore, the student is correct in thinking the following truth values are returned:

true
true



Short Description

Reference Variable Assignment - String (CC Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/AssignmentCopiesObject/

Question Text

Trace the execution of this program. What are the final values of variables a, b, and c?

```
public class Example{
   public static void main(String[] args){
      String a = "apple";
      String b = "banana";
      String c = "orange";
      a = b;
      b = c;
      c = "pineapple";
   }
}
```

Hint

Reference variables hold the address for the String literal stored in heap memory.

Answer 1 (Correct)

a -> "banana"; b -> "orange"; c -> "pineapple"

The reference to each string literal is swapped from b to a, c to b, and then c is set to "pineapple".

Feedback 1

You chose the correct option

Answer 2

a -> "banana"; b -> "orange"; c -> "pineapple"

All three variables are set to the value.

Feedback 2

You chose the incorrect option

Answer 3

a -> "pineapple"; b -> "pineapple"; c -> "pineapple"

All three variables are set to the value.

Feedback 3

You chose the incorrect option

Answer 4

a -> "pineapple"; b -> "pineapple"; c -> "pineapple"

The reference to each string literal is swapped from b to a, c to b, and then c is set to "pineapple".

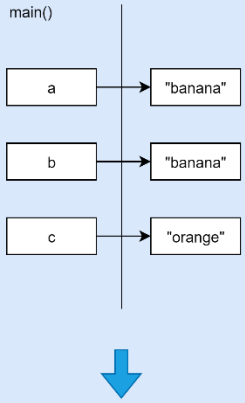
Feedback 4

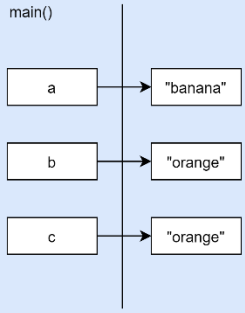
You chose the incorrect option

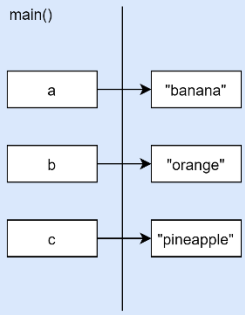
Overall Feedback

Valid Conception

On line 7, a is set to have the same value that is saved in b, therefore a = "banana". On line 8, b is set to have the same value as c, therefore b = "orange". On line 9, c is set the be a new value, therefore c = "pineapple".







Short Description

Reference Variable Assignment - String (LT Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/AssignmentCopiesObject/

Question Text

Trace the execution of this program. A student claims that the final values of a, b, and c are all "pineapple". Is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      String a = "apple";
      String b = "banana";
      String c = "orange";
      a = b;
      b = c;
      c = "pineapple";
   }
}
```

Hint

Reference variables hold the address for the String literal stored in heap memory.

Answer 1 (Correct)

a -> "banana"; b -> "orange"; c -> "pineapple"

The reference to each string literal is swapped from b to a, c to b, and then c is set to "pineapple".

Feedback 1

The student is therefore incorrect.

Answer 2

a -> "banana"; b -> "orange"; c -> "pineapple"

All three variables are set to the value.

Feedback 2

The student may still be confused.

Answer 3

a -> "pineapple"; b -> "pineapple"; c -> "pineapple"

All three variables are set to the value.

Feedback 3

The student may still be confused.

Answer 4

a -> "pineapple"; b -> "pineapple"; c -> "pineapple"

The reference to each string literal is swapped from b to a, c to b, and then c is set to "pineapple".

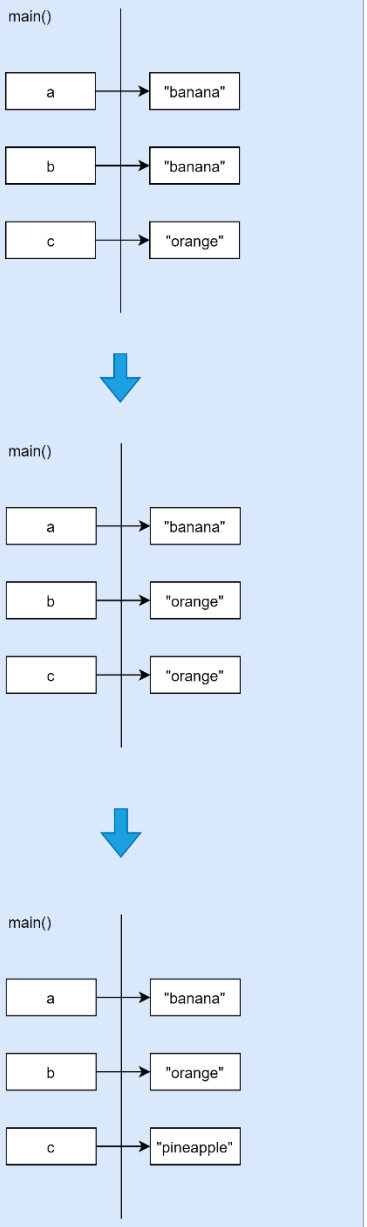
Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

On line 7, a is set to have the same value that is saved in b, therefore a = "banana". On line 8, b is set to have the same value as c, therefore b = "orange". On line 9, c is set the be a new value, therefore c = "pineapple". All three variables have different values, therefore the student is incorrect



Short Description

nulls Object (CC Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/NullIsObject/

Question Text

Trace the execution of this program. What are the values printed to the console?

```
public class Example {
   public static void main(String[] args) {
      String a = null;
      String b = null;
      String c = "null";
      System.out.println(a == b);
      System.out.println(a == c);
      System.out.println(a.equals(b));
   }
}
```

Hint

null is a reference pointing to no object.

Answer 1 (Correct)

true

false

Runtime exception error

There is an address to a null object allocated to the reference variable. Therefore, the comparison of the address returns true, then false as a string literal is assigned to a different location. Then, a runtime error is thrown due to a null pointer exception.

Feedback 1

You chose the correct option

Answer 2

true

false

Compilation error

You can only pass a String literal to the .equals method. It cannot reconcile the data type.

Feedback 2

You chose the incorrect option

Answer 3

Compilation error

There is no address allocated to the reference variable. Therefore the compilation error is thrown for the first comparison with the equality relational operator.

Feedback 3

You chose the incorrect option

Answer 4

Runtime exception error

A runtime error is thrown due to a null pointer exception.

Feedback 4

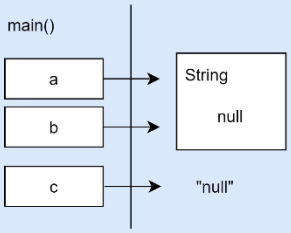
You chose the incorrect option

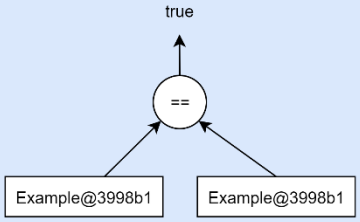
Overall Feedback

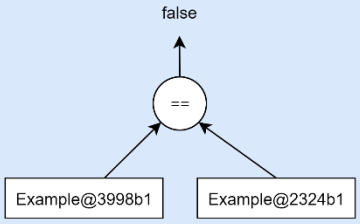
Valid Conception

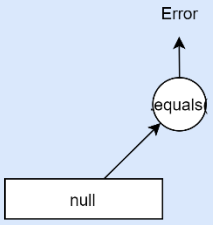
There is an address to a null object allocated to the reference variable. Therefore, the comparison of the address returns true, then false as a string literal is assigned to a different location. Then, a runtime error is thrown due to a null pointer exception.

true
false
Error - NullPointerException









Short Description

nulls Object (LT Lecture 5 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/NullIsObject/

Question Text

Trace the execution of this program. A student claims that a compilation error is thrown. Is this correct? Justify your answer.

```
public class Example {
   public static void main(String[] args) {
      String a = null;
      String b = null;
      String c = "null";
      System.out.println(a == b);
      System.out.println(a == c);
      System.out.println(a.equals(b));
   }
}
```

Hint

null is a reference pointing to no object.

Answer 1 (Correct)

true

false

Runtime exception error

There is an address to a null object allocated to the reference variable. Therefore, the comparison of the address returns true, then false as a string literal is assigned to a different location. Then, a runtime error is thrown due to a null pointer exception.

Feedback 1

The student is therefore incorrect.

Answer 2

true

false

Compilation error

You can only pass a String literal to the .equals method. It cannot reconcile the data type.

Feedback 2

The student may be still confused.

Answer 3

Compilation error

There is no address allocated to the reference variable. Therefore the compilation error is thrown for the first comparison with the equality relational operator.

Feedback 3

The student may be still confused.

Answer 4

Runtime exception error

A runtime error is thrown due to a null pointer exception.

Feedback 4

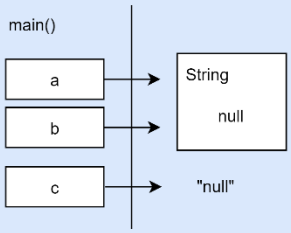
The student may be still confused.

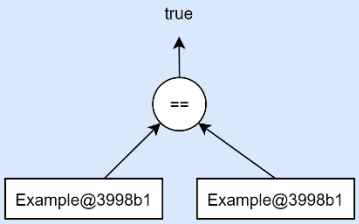
Overall Feedback

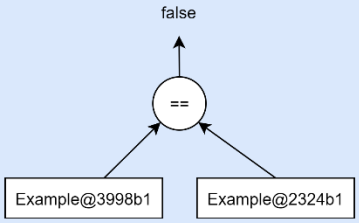
Valid Conception

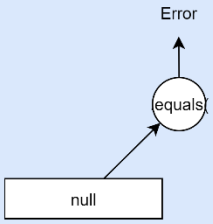
There is an address to a null object allocated to the reference variable. Therefore, the comparison of the address returns true, then false as a string literal is assigned to a different location. Then, a runtime error is thrown due to a null pointer exception. The student is therefore incorrect.

true
false
Error - NullPointerException









Short Description

Local Primitive Variable - Scope (CC Lecture 6 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. What value is printed to the console?

```
public class Example{
   public static void main(String[] args){
      int x = 1;
      {
         int y = x + 1;
      }
      System.out.println(x + y);
   }
}
```

Hint

The scope of a variable is from the point of its initialization to the closing curly brace that marks the end of the block of code.

Answer 1 (Correct)

No value is printed, a compilation error is thrown

The variable y has been dropped from stack memory. Therefore, no value is printed because y cannot be resolved to a variable.

Feedback 1

You chose the correct option

Answer 2

No value is printed, a compilation error is thrown

The variable y is still in stack memory. The value cannot be accessed however and the error is thrown.

Feedback 2

You chose the incorrect option

Answer 3

3

The variable y is still in stack memory. The value can be accessed and the returned value of the arithmetic operation is printed to the console.

Feedback 3

You chose the incorrect option

Answer 4

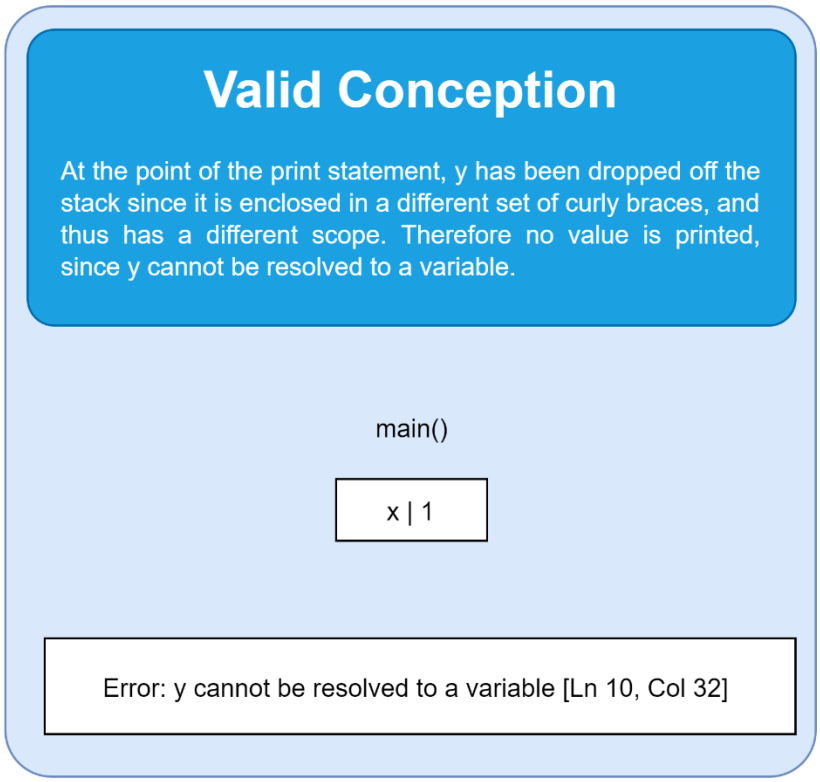
1

The variable y is still in stack memory. The default value of 0 is accessed and read from memory and the returned value of the arithmetic operation is printed to the console.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

Local Primitive Variable - Scope (LT Lecture 6 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. A student claims that the output of this program is 3. Is the student correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      int x = 1;
      {
         int y = x + 1;
      }
      System.out.println(x + y);
   }
}
```

Hint

The scope of a variable is from the point of its initialization to the closing curly brace that marks the end of the block of code.

Answer 1 (Correct)

No value is printed, a compilation error is thrown

The variable y has been dropped from stack memory. Therefore, no value is printed because y cannot be resolved to a variable.

Feedback 1

The student is therefore incorrect.

Answer 2

No value is printed, a compilation error is thrown

The variable y is still in stack memory. The value cannot be accessed however and the error is thrown.

Feedback 2

The student may still be confused.

Answer 3

3

The variable y is still in stack memory. The value can be accessed and the returned value of the arithmetic operation is printed to the console.

Feedback 3

The student may still be confused.

Answer 4

1

The variable y is still in stack memory. The default value of 0 is accessed and read from memory and the returned value of the arithmetic operation is printed to the console.

Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

At the point of the print statement, y has been dropped off the stack since it is enclosed in a different set of curly braces, and thus has a different scope. Therefore the student is incorrect since this code prints nothing.

main()

x | 1

Error: y cannot be resolved to a variable [Ln 10, Col 32]

Short Description

If Is Loop (CC Lecture 6 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/IfIsLoop/

Question Text

Trace the execution of this program. What is the value printed to the console?

```
public class Example{
   public static void main(String[] args){
      int a = 5;
      if(a > 0){
         a--;
      }
      System.out.println(a);
   }
}
```

Hint

A selection statement allows for branching scenarios - a block of code is executed, but only when a condition is met.

Answer 1 (Correct)

4

Because a is set to 5 and is greater than 0, the evaluation of the condition returns true. The value of a is decremented by one.

Feedback 1

You chose the correct option

Answer 2

0

The value of a is decremented by one until the condition is met. Therefore, only when a is set to 0 does the condition return false, and the value is printed to the console.

Feedback 2

You chose the incorrect option

Answer 3

5

Because a is set to 5 and is greater than 0, the evaluation of the condition returns false. The value of a is printed to the console.

Feedback 3

You chose the incorrect option

Answer 4

5

Because a is set to 5 and is greater than 0, the evaluation of the condition returns false. The original value of a is returned from the prefix decrement operation, and the value is printed to the console.

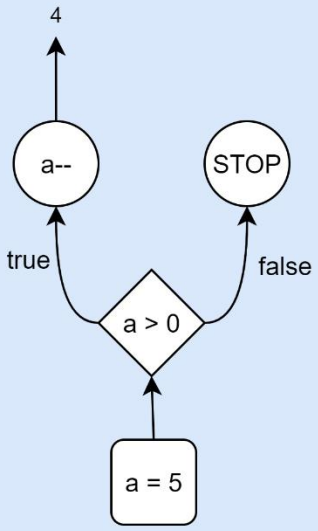
Feedback 4

You chose the incorrect option

Overall Feedback

Valid Conception

The if statement checks to see if a is greater than 5. Because a is greater than 5, it is decremented by one. Therefor the value printed is 4.



Short Description

If Is Loop (LT Lecture 6 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/IfIsLoop/

Question Text

Trace the execution of this program. A student claims that the value printed ot the console is 0. Is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      int a = 5;
      if(a > 0){
         a--;
      }
      System.out.println(a);
   }
}
```

Hint

A selection statement allows for branching scenarios - a block of code is executed, but only when a condition is met.

Answer 1 (Correct)

4

Because a is set to 5 and is greater than 0, the evaluation of the condition returns true. The value of a is decremented by one.

Feedback 1

The student is therefore incorrect

Answer 2

0

The value of a is decremented by one until the condition is met. Therefore, only when a is set to 0 does the condition return false, and the value is printed to the console.

Feedback 2

The student may still be confused.

Answer 3

5

Because a is set to 5 and is greater than 0, the evaluation of the condition returns false. The value of a is printed to the console.

Feedback 3

The student may still be confused.

Answer 4

5

Because a is set to 5 and is greater than 0, the evaluation of the condition returns false. The original value of a is returned from the prefix decrement operation, and the value is printed to the console.

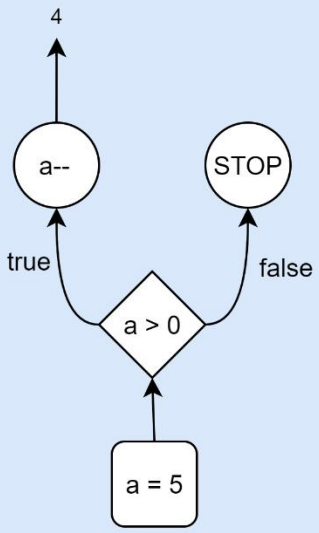
Feedback 4

The student may still be confused.

Overall Feedback

Valid Conception

The if statement checks to see if a is greater than 5. Because a is greater than 5, it is decremented by one. Therefor the value printed is 4 and the student is incorrect



Short Description

no Short Circuit (CC Lecture 7 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/NoShortCircuit/

Question Text

Trace the execution of this program. What are the values printed to the console?

```
public class Example {
   public static void main(String[] args) {
      int a = 1;
      int b = 2;
      int c = 3;
      if(a == 1 || b == 1 && !c == 3){
         System.out.println(a + b + c);
      }
   }
}
```

Hint

The NOT operator inverses boolean values.

Answer 1 (Correct)

Compilation error

The NOT operator can only be applied to Boolean values, not integers. A compilation error is thrown.

Feedback 1

You chose the correct option

Answer 2

Runtime exception

The NOT operator can only be applied to Boolean values, not integers. A runtime exception is thrown following the evaluation of the first operand of the OR operator (i.e., a == 1).

Feedback 2

You chose the incorrect option

Answer 3

Runtime exception

The NOT operator can only be applied to Boolean values, not integers. A runtime exception is thrown following the evaluation of the first and second operand of the OR operator (i.e., a == 1, then b == 1).

Feedback 3

You chose the incorrect option

Answer 4

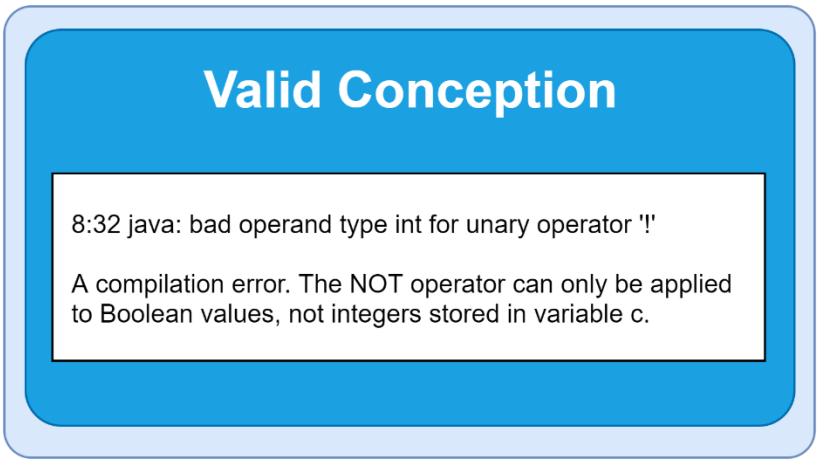
No value

The AND operator evaluates for operand values true and false, and returns false. Therefore, the print statement is not reached, and nothing appears in the console.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

no Short Circuit (LT Lecture 7 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/NoShortCircuit/

Question Text

Trace the execution of this program. A student claims that no value is printed to the console. Is this correct? Justify your answer.

```
public class Example {
   public static void main(String[] args) {
      int a = 1;
      int b = 2;
      int c = 3;
      if(a == 1 || b == 1 && !c == 3){
         System.out.println(a + b + c);
      }
   }
}
```

Hint

The NOT operator inverses boolean values.

Answer 1 (Correct)

Compilation error

The NOT operator can only be applied to Boolean values, not integers. A compilation error is thrown.

Feedback 1

The student is therefore incorrect.

Answer 2

Runtime exception

The NOT operator can only be applied to Boolean values, not integers. A runtime exception is thrown following the evaluation of the first operand of the OR operator (i.e., a == 1).

Feedback 2

The student may still be confused.

Answer 3

Runtime exception

The NOT operator can only be applied to Boolean values, not integers. A runtime exception is thrown following the evaluation of the first and second operand of the OR operator (i.e., a == 1, then b == 1).

Feedback 3

The student may still be confused.

Answer 4

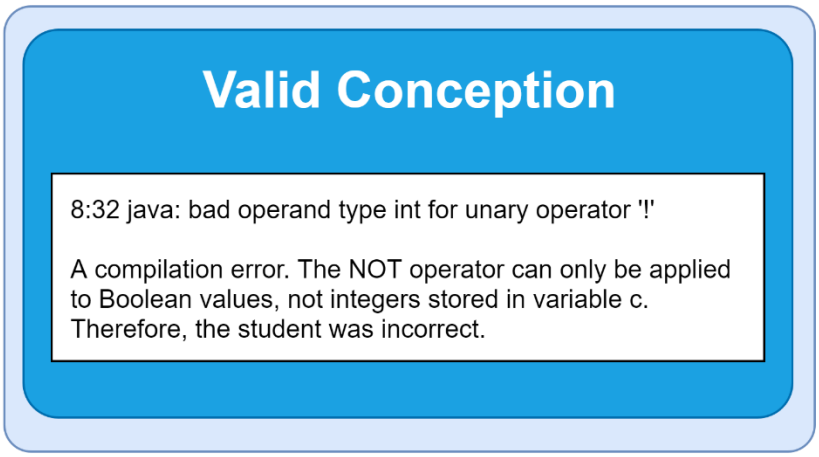
No value

The AND operator evaluates for operand values true and false, and returns false. Therefore, the print statement is not reached, and nothing appears in the console.

Feedback 4

The student may still be confused.

Overall Feedback



Short Description

Reference Variable Comparison - Arrays (CC Lecture 10 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program and predict the output. What values are printed to the console?

```
import java.util.Arrays;

public class Example{
   public static void main(String[] args){
      String[] a = {"apple", "banana", "orange"};
      String[] b = {"apple", "banana", "orange"};
      System.out.println(a == b);
      System.out.println(Arrays.equals(a, b));
   }
}
```

Hint

Both variables are of reference type holding the address where the array is located on heap memory.

Answer 1 (Correct)

false

true

The first statement is checking the equality of the memory locations of a and b, whereas the second is checking the values inside a and b.

Feedback 1

You chose the correct option

Answer 2

false

true

The first statement is checking the values inside a and b, whereas the second is checking the memory locations of a and b.

Feedback 2

You chose the incorrect option

Answer 3

true

true

Both arrays are located on the same address in heap memory.

Feedback 3

You chose the incorrect option

Answer 4

true

true

Both arrays have similar element values.

Feedback 4

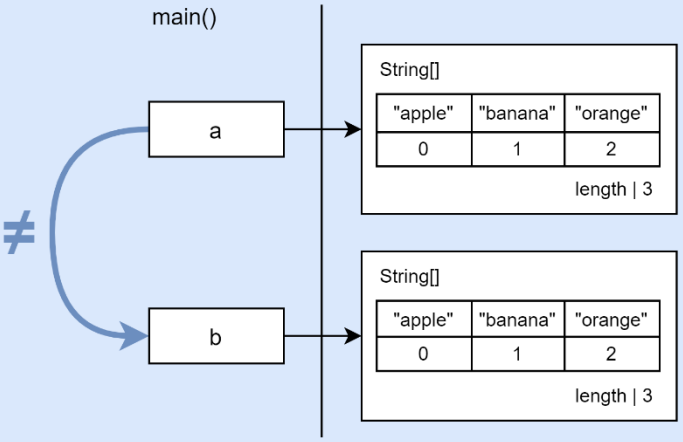
You chose the incorrect option

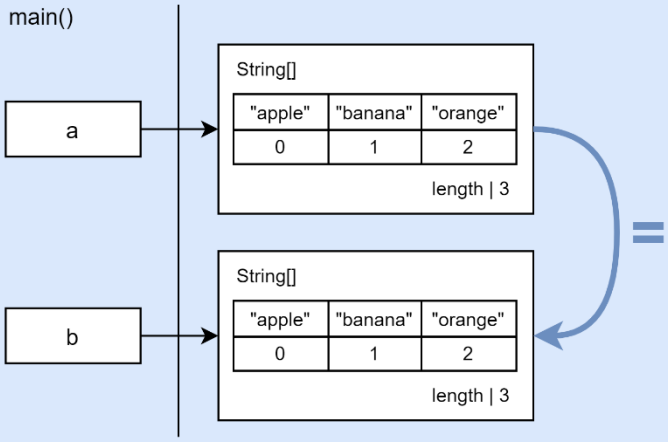
Overall Feedback

Valid Conception

Although the two statements are checking equality, they are checking equality of different things. The first statement is checking the equality of the memory locations of a and b, whereas the second is checking the values inside a and b. Therefore the values printed are:

false
true





Short Description

Reference Variable Comparison - Arrays (LT Lecture 10 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

A student claims that the Boolean values printed in the console are false true. Explain whether or not the student is correct, and justify your answer.

```
import java.util.Arrays;

public class Example{

   public static void main(String[] args){

      String[] a = {"apple", "banana", "orange"};

      String[] b = {"apple", "banana", "orange"};

      System.out.println(a == b);

      System.out.println(Arrays.equals(a, b));

   }

}
```

Hint

Both variables are of reference type holding the address where the array is located on heap memory.

Answer 1 (Correct)

false

true

The first statement is checking the equality of the memory locations of a and b, whereas the second is checking the values inside a and b.

Feedback 1

The student chose the correct option.

Answer 2

false

true

The first statement is checking the values inside a and b, whereas the second is checking the memory locations of a and b.

Feedback 2

The student may still be confused.

Answer 3

true

true

Both arrays are located on the same address in heap memory.

Feedback 3

The student may still be confused.

Answer 4

true

true

Both arrays have similar element values.

Feedback 4

The student may still be confused.

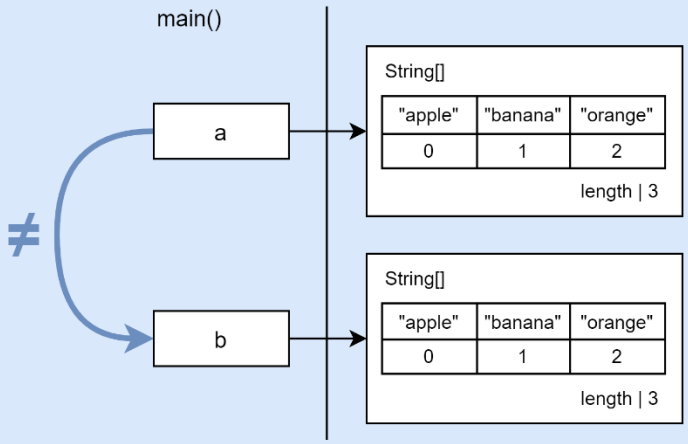
Overall Feedback

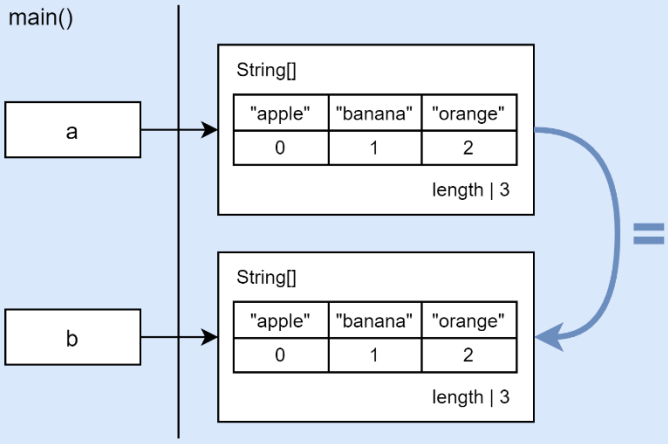
Valid Conception

Although the two statements are checking equality, they are checking equality of different things. The first statement is checking the equality of the memory locations of a and b, whereas the second is checking the values inside a and b. Therefore the values printed are:

false
true

and the student is correct.





Short Description

Reference Variable Assignment - Arrays (CC Lecture 10 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. What are the final values of arrA and arrB?

```
public class Example{
   public static void main(String[] args){
      int[] arrA = {42, -10, 29};
      int[] arrB = {3, 77, -51};
      arrB = arrA;
      arrA[2] = 99;
      arrB[0] = 67;
   }
}
```

Hint

Both variables are of reference type holding the address where the array is located on heap memory.

Answer 1 (Correct)

arrA is set to {67, -10, 99}

arrB is set to {67, -10, 99}

Since both arrA and arrB both hold a reference to the same array stored in heap memory, the assignment operations change element values in the same array.

Feedback 1

You chose the correct option

Answer 2

arrA is set to {42, -10, 99}

arrB is set to {67, -10, 29}

Since both ArrB is set to the same values as arrA, the arrA is then updated to 99 at index 2, and arrB is set to 67 at index 0.

Feedback 2

You chose the incorrect option

Answer 3

arrA is set to {67, -10, 99}

arrB is set to {67, -10, 99}

Since both ArrB is set to the same values as arrA, the arrA is then updated to 99 at index 2, and arrB is set to 67 at index 0.

Feedback 3

You chose the incorrect option

Answer 4

arrA is set to {42, -10, 99}

arrB is set to {67, -10, 29}

Since both arrA and arrB both hold a reference to the same array stored in heap memory, the assignment operations change element values in the same array.

Feedback 4

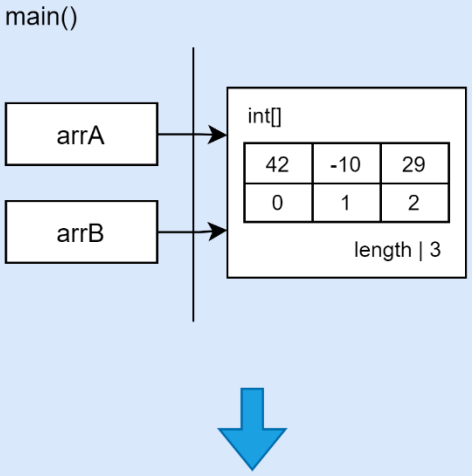
You chose the incorrect option

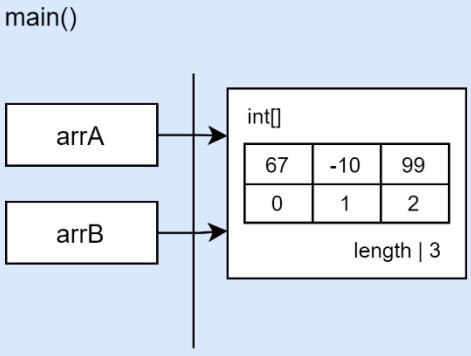
Overall Feedback

Valid Conception

On line 6, arrB is set to share the same memory location as arrB. Since both arrA and arrB now point to the same location, the two following modifications apply to the same array, so the final values are:

arrA = {67, -10, 99}
arrB = {67, -10, 99}





Short Description

Reference Variable Assignment - Arrays (LT Lecture 10 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. A student claims that arrA = {42, -10, 99} and arrB = {67, -10, 29} at the end of the program. Are these values correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      int[] arrA = {42, -10, 29};
      int[] arrB = {3, 77, -51};
      arrB = arrA;
      arrA[2] = 99;
      arrB[0] = 67;
   }
}
```

Hint

Both variables are of reference type holding the address where the array is located on heap memory.

Answer 1 (Correct)

arrA is set to {67, -10, 99}

arrB is set to {67, -10, 99}

Since both arrA and arrB both hold a reference to the same array stored in heap memory, the assignment operations change element values in the same array.

Feedback 1

The student is therefore incorrect.

Answer 2

arrA is set to {42, -10, 99}

arrB is set to {67, -10, 29}

Since both ArrB is set to the same values as arrA, the arrA is then updated to 99 at index 2, and arrB is set to 67 at index 0.

Feedback 2

The student may still be confused.

Answer 3

arrA is set to {67, -10, 99}

arrB is set to {67, -10, 99}

Since both ArrB is set to the same values as arrA, the arrA is then updated to 99 at index 2, and arrB is set to 67 at index 0.

Feedback 3

The student may still be confused.

Answer 4

arrA is set to {42, -10, 99}

arrB is set to {67, -10, 29}

Since both arrA and arrB both hold a reference to the same array stored in heap memory, the assignment operations change element values in the same array.

Feedback 4

The student may still be confused.

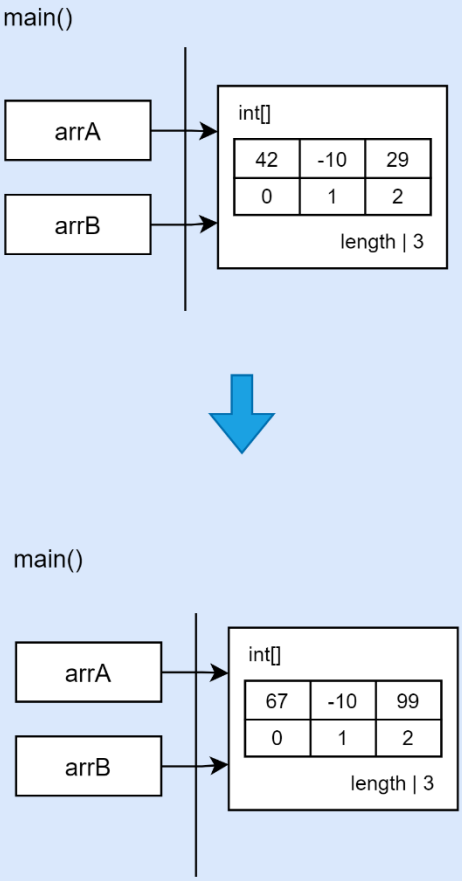
Overall Feedback

Valid Conception

On line 6, arrB is set to share the same memory location as arrB. Since both arrA and arrB now point to the same location, the two following modifications apply to the same array, so the final values are:

arrA = {67, -10, 99}
arrB = {67, -10, 99}

and the student is incorrect



Short Description

Array Grows (CC Lecture 10 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArraysGrow/

Question Text

Trace the execution of this program. What are the element values of the array?

```
public class Example{
   public static void main(String[] args){
      int[] arr = new int[5];
      arr[0] = 62;
      arr[1] = 11;
      arr[2] = 38;
      arr[3] = 99;
      arr[4] = 17;
      arr[5] = 3;
   }
}
```

Hint

The length of the array is defined when initialized.

Answer 1 (Correct)

A value is out of bounds of the array, a runtime exception is thrown.

The length of the array is of 5 elements, therefore the sixth value cannot be set.

Feedback 1

You chose the correct option

Answer 2

A value is out of bounds of the array, a runtime exception is thrown.

The length of the array is of 6 elements, therefore the sixth value can be set.

Feedback 2

You chose the incorrect option

Answer 3

The array is set to {62, 11, 38, 99, 17, 3}

The length of the array is of 5 elements, therefore the sixth value cannot be set.

Feedback 3

You chose the incorrect option

Answer 4

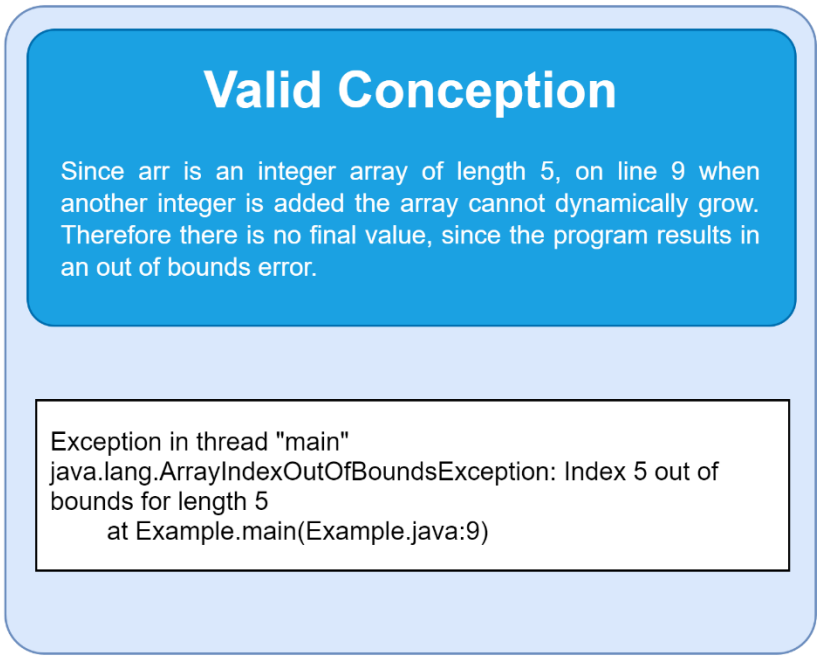
The array is set to {62, 11, 38, 99, 17, 3}

The length of the array is of 6 elements, therefore the sixth value can be set.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

Array Grows (LT Lecture 10 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ArraysGrow/

Question Text

Trace the execution of this program. A student claims that the values of the array are {62, 11, 38, 99, 17, 3}. Is this correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      int[] arr = new int[5];
      arr[0] = 62;
      arr[1] = 11;
      arr[2] = 38;
      arr[3] = 99;
      arr[4] = 17;
      arr[5] = 3;
   }
}
```

Hint

The length of the array is defined when initialized.

Answer 1 (Correct)

A value is out of bounds of the array, a runtime exception is thrown.

The length of the array is of 5 elements, therefore the sixth value cannot be set.

Feedback 1

The student is therefore incorrect.

Answer 2

A value is out of bounds of the array, a runtime exception is thrown.

The length of the array is of 6 elements, therefore the sixth value can be set.

Feedback 2

The student may still be confused.

Answer 3

The array is set to {62, 11, 38, 99, 17, 3}

The length of the array is of 5 elements, therefore the sixth value cannot be set.

Feedback 3

The student may still be confused.

Answer 4

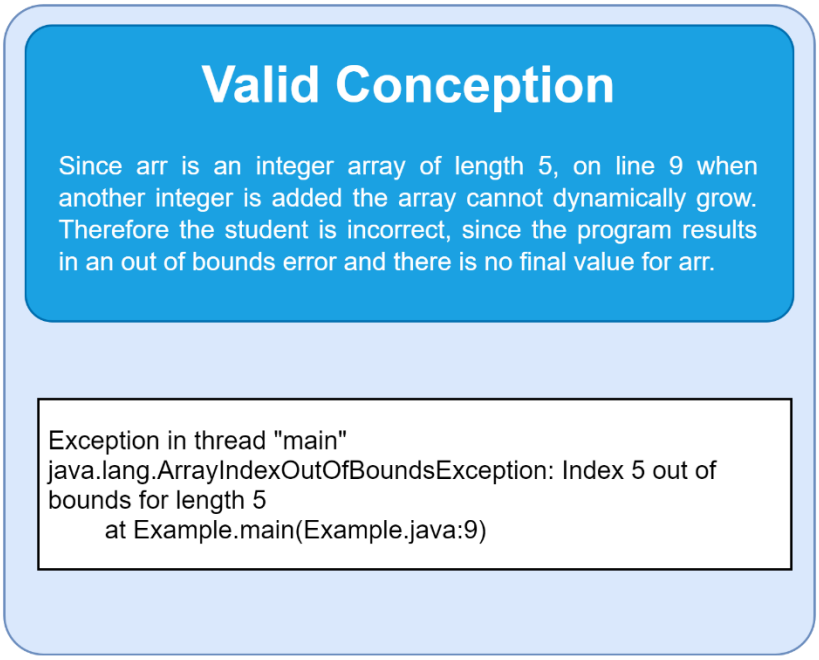
The array is set to {62, 11, 38, 99, 17, 3}

The length of the array is of 6 elements, therefore the sixth value can be set.

Feedback 4

The student may still be confused.

Overall Feedback



Short Description

Local Primitive Method - Scope (CC Lecture 14 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. What is the final value of variable x?

```
public class Example{
   public static void main(String[] args){
      int x = 44;
      method(x);
      System.out.println(x);
   }
   public static void method(int x){
      x = 15;
   }
}
```

Hint

Notice that the method is of return type void - no value is returned to the caller of the method.

Answer 1 (Correct)

44

Since x is initially set to 44, and its scope is local to the main method. The call of the method returns no value.

Feedback 1

You chose the correct option

Answer 2

44

Since x is initially set to 44, but then updated to 15 within the method. Reference variable values have to be returned to the caller of the method in order for the value to be updated.

Feedback 2

You chose the incorrect option

Answer 3

15

Since x is initially set to 44, but then updated to 15 within the method.

Feedback 3

You chose the incorrect option

Answer 4

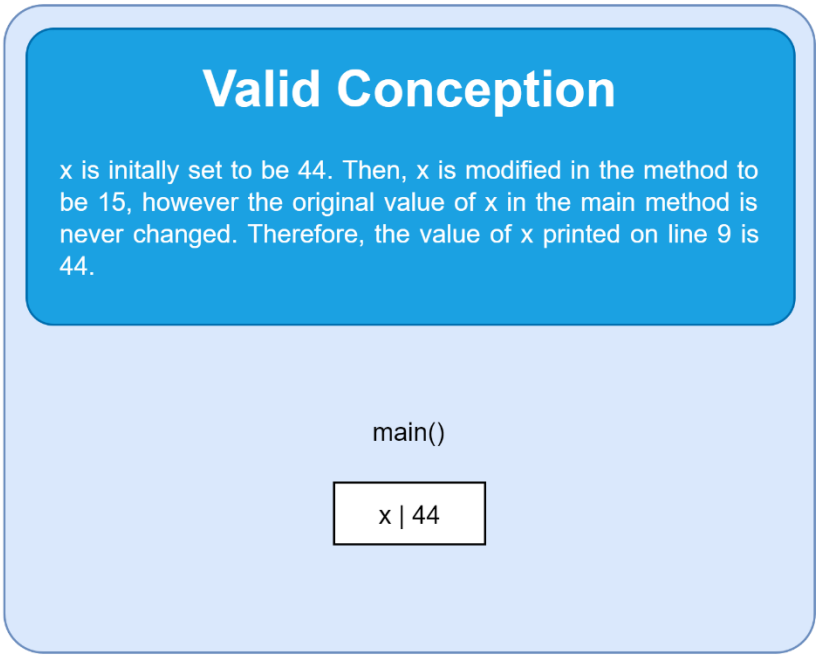
15

Since x is initially set to 44, but then updated to 15 within the method. Reference variable values have to be returned to the caller of the method in order for the value to be updated.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

Local Primitive Method - Scope (LT Lecture 14 - Memory Diagram)

Reference

Colleen M. Lewis. 2021. Physical Java Memory Models: A Notional Machine. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432477

URL: https://dl.acm.org/doi/pdf/10.1145/3408877.3432477

Question Text

Trace the execution of this program. A student claims that the output of this program is 15. Is the student correct? Justify your answer.

```
public class Example{
   public static void main(String[] args){
      int x = 44;
      method(x);
      System.out.println(x);
   }
   public static void method(int x){
      x = 15;
   }
}
```

Hint

Notice that the method is of return type void - no value is returned to the caller of the method.

Answer 1 (Correct)

44

Since x is initially set to 44, and its scope is local to the main method. The call of the method returns no value.

Feedback 1

The student is therefore incorrect.

Answer 2

44

Since x is initially set to 44, but then updated to 15 within the method. Reference variable values have to be returned to the caller of the method in order for the value to be updated.

Feedback 2

The student may still be confused.

Answer 3

15

Since x is initially set to 44, but then updated to 15 within the method.

Feedback 3

The student may still be confused.

Answer 4

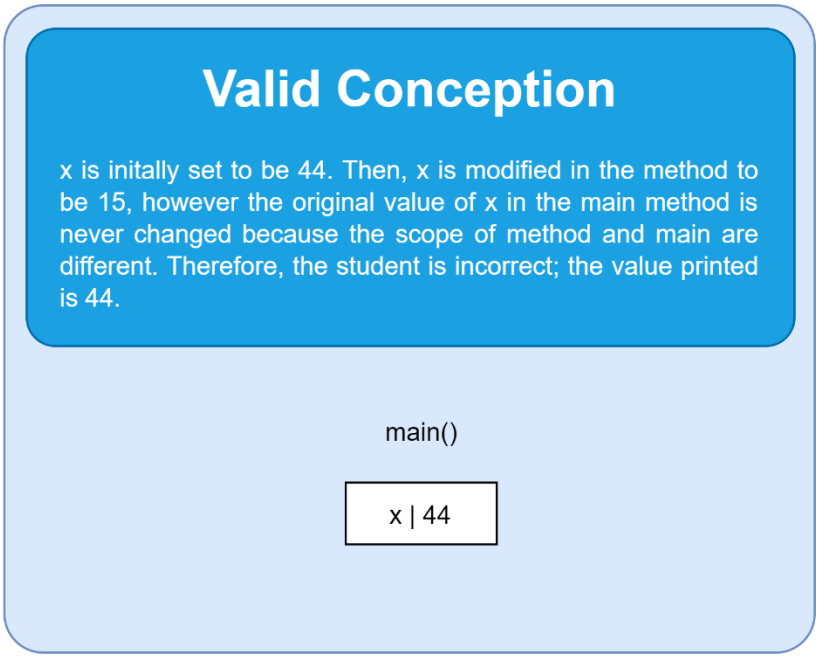
15

Since x is initially set to 44, but then updated to 15 within the method. Reference variable values have to be returned to the caller of the method in order for the value to be updated.

Feedback 4

The student may still be confused.

Overall Feedback



Short Description

this Exists in Static Method (CC Lecture 17 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ThisExistsInStaticMethod/

Question Text

Trace the execution of this program. What is the final value of the instance variable c?

```
public class Example {
   private int c;
   public static void main(String[] args) {
      int a = 1;
      Example b = new Example();
      this.c = a;
      System.out.println(b.c);
   }
}
```

Hint

this is used as a reference to objects and may be used in methods that perform operations on objects.

Answer 1 (Correct)

Compilation error

this cannot be reference from a static method.

Feedback 1

You chose the correct option

Answer 2

Compilation error

this is like any other local variable, and can be used in a static method. Therefore, this.c is set to a, a value of 1, and prints the same value to the console.

Feedback 2

You chose the incorrect option

Answer 3

1

this is like any other local variable, and can be used in a static method. Therefore, this.c is set to a, a value of 1, and prints the same value to the console.

Feedback 3

You chose the incorrect option

Answer 4

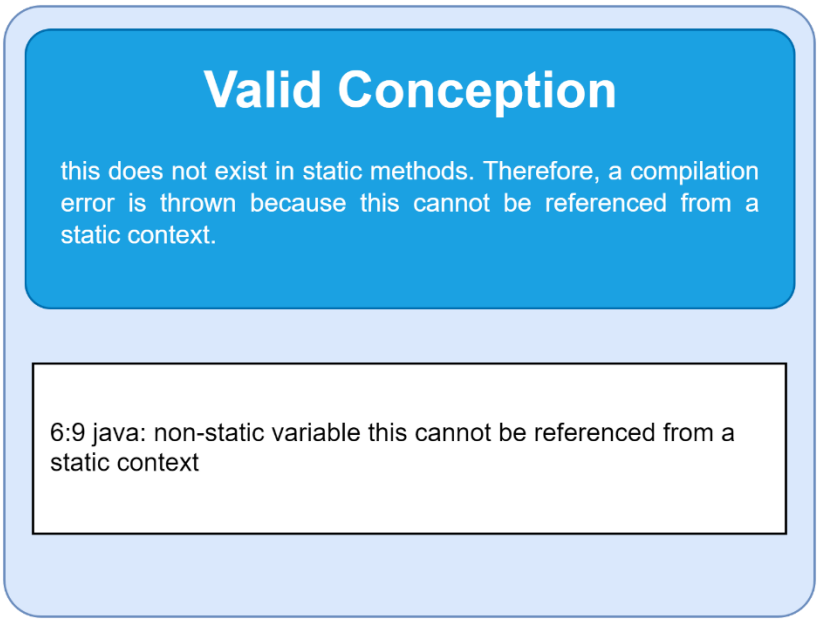
1

this cannot be reference from a static method.

Feedback 4

You chose the incorrect option

Overall Feedback



Short Description

this Exists in Static Method (LT Lecture 17 - Memory Diagram)

Reference

Luca Chiodini, Igor Moreno Santos, Andrea Gallidabino, Anya Tafliovich, André L. Santos, Matthias Hauswirth. A Curated Inventory of Programming Language Misconceptions. Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '21), June 26-July 1, 2021, Virtual Event, Germany. 10.1145/3430665.3456343

URL: https://progmiscon.org/misconceptions/Java/ThisExistsInStaticMethod/

Question Text

Trace the execution of this program. A student claims that 1 is printed to the console. Is this correct? Justify your answer.

```
public class Example {
   private int c;
   public static void main(String[] args) {
      int a = 1;
      Example b = new Example();
      this.c = a;
      System.out.println(b.c);
   }
}
```

Hint

this is used as a reference to objects and may be used in methods that perform operations on objects.

Answer 1 (Correct)

Compilation error

this cannot be reference from a static method.

Feedback 1

The student is therefore incorrect

Answer 2

Compilation error

this is like any other local variable, and can be used in a static method. Therefore, this.c is set to a, a value of 1, and prints the same value to the console.

Feedback 2

The student may still be confused.

Answer 3

1

this is like any other local variable, and can be used in a static method. Therefore, this.c is set to a, a value of 1, and prints the same value to the console.

Feedback 3

The student may still be confused.

Answer 4

1

this cannot be reference from a static method.

Feedback 4

The student may still be confused.

Overall Feedback