Cognizant Technology Solutions

Wrapper Classes Exercise

CATP Java Team
5/20/2012

For The Associates:

The documents details two flavors of problem statements

- **Statement # 1:** Few problem solutions have been provided for associates should analyze the program and write down the program output. This will enhance the analyzing skills of associates and also understand "why" part of java programming feature. The associates can then try running the program in eclipse and check if the output with what they have written.
- **Stamen # 2:** There are some problem statements provided similar to the final assessment and associates need to solve it. This will enhance the programming skills of the associates.
  IMPORTANT: These exercises will gear you up for the core java assessment so please develop/analyze the exercise independently. In case you are stuck up reach out to the trainers.

Exercises:

1. What is the output of compiling and running the following code?

```
class Test {
public static void main(String arg[]) {
Integer x = 1000;
long y = 1000;
Integer z = 1000;
System.out.println(x==y);
System.out.println(x==z);
}
}
```

2. What is the output?

```
class Tester {
public static void main(String[] args) {
Integer x = 0;
Integer y = 0;
for(Short z = 0; z < 5; z++)
if((++x > 2) || (++y > 2))
x++;
System.out.println(x + " " + y);
}
}
```

3. What is the output?

```
class Titanic {
public static void main(String[] args) {
Boolean b1 = true;
boolean b2 = false;
boolean b3 = true;
if((b1 & b2) | (b2 & b3) & b3)
System.out.print("alpha ");
if((b1 = false) | (b1 & b3) | (b1 | b2))
System.out.print("beta ");
} }
```

4. What is the output?

```
class Tester {

int doX(Long x, Long y) { return 1; }

int doX(long x, long y) { return 2; }

int doX(Integer x, Integer y) { return 3; }

int doX(Number n, Number m) { return 4; }

public static void main(String[] args) {

new Tester().go();

}

void go() {

short s = 7;

System.out.print(doX(s,s) + " ");

System.out.println(doX(7,7));

}

}
```

5. Given the code. What is the result?

```
public class TryMe {
  Integer A;
     int a;
    public TryMe(int a) {
      this.a = A + a;
     System.out.print(this.a);
  }
   public static void main(String args[]) {
     Integer A = new Integer(1);
      TryMe t = new TryMe(A);
   }
}
```

6. Create a class that performs the conversion between data type values.

Class Name	TypeConvertor
Method Name	binaryToLong
Method Description	Converts the binary number to Long object
Argument	String binaryNumber
Return Type	Long – Long Equivalent of the binaryNumber
Logic	Convert the binary to Long using the appropriate APIs
Method Name	stringToInteger
Method Description	Converts the string to an integer
Argument	String integerValue
Return Type	int – Integer Values
Logic	Convert the string to integer using the appropriate APIs

7. Create a class that finds the smallest of an array of numbers.

Class Name	EvenSum
Method Name	getSmallest
Method Description	Finds the smallest member of an integer array
Argument	int [ ]elements
Return Type	int – Smallest number in the array
Logic	Iterate through the array and find the smallest element in the array.

8. Create a class that finds the sum of elements in an int array

Class Name	ArrayManipulator
Method Name	getArraySum
Method Description	Finds the sum of elements in an int array
Argument	int [ ]elements
Return Type	int – sum of the elements in the array
Logic	Iterate through the elements in the array and find the sum of the elements

9. Create a class that contains method to accept an array and convert all the even members to the next odd (even+1) and all the odd numbers to the previous even (odd-1).

Class Name	ArrayManipulator
Method Name	changeNumber
Method Description	Changes even to odd and odd to even
Argument	int [ ]elements
Return Type	int[ ] – The resulting array after the process
Logic	Iterate through the array, if an even element is found change it to the next odd and if an odd number is found change it to the previous even. For example if the array is {5,6,8,9} Output : {6,5,7,10}

10. Create a class that contains method which can remove the specified character from the given character array

Class Name	ArrayManipulator
Method Name	removeCharacter
Method Description	Removes the specified character from the character array
Argument	char[ ], char
Return Type	char[ ] – The resulting array after the process
Logic	Iterate through the array, remove the entered character if found and return the resultant array.

11. Create a class that contains method capable of modifying in an array such a way that the new array contains elements which are the product of two neighboring columns in the original array. Last element can be left as it is.

Class Name	ArrayManipulator
Method Name	changeArrau
Method Description	Changes each value of each column as the product of two neighboring columns
Argument	int []
Return Type	int[ ] – The resulting array after the process
Logic	Iterate through the array and set the values of each column as the product of two neighboring columns. Last element can be left as it is. For Example If the input array is {1,2,3,4} The resulting array should be {2,6,12,4}