**Worksheet 9 - Arrays**

1. For each of the following array declaration, mark whether it is valid or invalid. (Invalid includes both compile-time and run-time errors).

|  |  |
| --- | --- |
| **Array Declaration** | **✓ OR 🗶** |
| int[] = new int[10]; |  |
| int i = new int[7]; |  |
| String[] s = new String[5]; |  |
| String[] s = new String[]; |  |
| String[] s = {“dog”,”bark”,”growl”}; |  |
| boolean[] flags = new boolean[3]; |  |

1. Consider the following array, and for each statement below, mark whether it is valid or invalid. (Invalid includes both compile-time and run-time errors).

int [] nums = {0,1,2,3,6,8};

|  |  |
| --- | --- |
| **Statement** | **✓ OR 🗶** |
| System.out.println(nums[0]); |  |
| System.out.println(nums[-1]); |  |
| nums[3] = 5; |  |
| nums[4] = nums[5]; |  |
| nums[6] = 10; |  |

1. Consider the code snippet below:

int[] scores = new int[3];

int sum = 0;

scores[0] = 8;

scores[1] = 5;

scores[2] = 7;

for (int i = 0; i < 3; i++)

sum += scores[i];

System.out.println(sum / scores.length);

What would the println statement output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Given the following:

int i = 3;

int [] arr = {10,15,7,20,11,2};

int x = -1;

int y = 100;

Show the values in x and y after executing each of the following:

1. x = arr[3]; x = \_\_\_\_\_\_\_\_\_
2. x = 2 \* arr[0]; x = \_\_\_\_\_\_\_\_\_
3. x = arr[1] + arr[2]; x = \_\_\_\_\_\_\_\_\_
4. x = arr[i] + i; x = \_\_\_\_\_\_\_\_\_

y = x + arr[5]; y = \_\_\_\_\_\_\_\_\_

1. x = arr[i] + 1; x = \_\_\_\_\_\_\_\_\_

y = arr[i+1]; y = \_\_\_\_\_\_\_\_\_

1. Given the following array:

int[] a = {-2, 4, 0, -5, 7, 9, -8, 5, 6, -3};

Write a loop to do each of the following:

1. Add 1 to every element of *a*.
2. Count the number of negative numbers in *a*.
3. Create a new array *b* which is the same size as *a*, and copy all elements from *a* into *b*.
4. Print the elements of *a* in reverse order.
5. Count the number of elements in *a* that have values between 0 and 10 inclusive.
6. Write a method named *findPosition* that accepts two parameters, an int named *key* and an array of int’s named *numbers*. The method must return the position of the key in the numbers array. Return the value -1 if key is not stored in the array.
7. Write a program that saves all combinations of cards in two separate arrays (numbers and suits), and then randomly generates a face and its suit. Example:

Number: 2

Suit: Spades

Display: 2 of Spades

The first 4 generated combinations should be saved in an array of Strings and then displayed one after the other.

**Help:**

Numbers: Ace, 2-9, Jack, Queen, King

Suits: Spades, Hearts, Clubs, Diamonds

1. Write a program that rolls a die for 10000 times. You should also create an array with 6 elements in which you should store the frequency of the occurrence of each face (1, 2, 3, etc.). Output the result.