CSE 215: Programming Language II Lab

Lab – 2

Lab Instructor: Tanzina Tazreen

Objective:

- To learn different kinds loops : for, while, do-while
- To learn to about array
- To learn more about 2D array

Loops allow us to perform repetitive tasks within a few blocks of code instead of doing them manually (e.g. copy pasting).
In Java, there are mainly three types of loops:
a. for loop
b. while loop
c. do…while loop

for

```
for(int i = 0; i < 10; i++){
// do something
}
```

while

```
while(condition){
// do something
}
```

Anything that can be done using a `for` loop, can also be done using a `while` loop

Nested loops:
Nested loops are simply loop within a loop.
However, the more loops you have within a nested loop, the more computationally expensive it becomes.

Syntax: Nested for loop

```
for (int i = 1; i <=5; i++) {
// outer loop body
for (int j = 1; j <=5; j++) {
// inner loop body
}
// outer loop body
}
```

Nested while loop

```
while(someCondition) {
// outer while loop body
while(someOtherCondition) {
// inner while loop body
}
// outer while loop body
}
```

An **Array** is a collection of **similar type** of elements which have a **contiguous memory location (Stack Memory Location)**.

- Java supports arrays of primitive data types, similar to C.
- Unlike C, Java also has support for arrays of the String datatype.
- Similar to C, Java arrays use 0-based indexing.

Declaring arrays:

```
<datatype>[] <array_identifier> = new datatype[size];
```

For example:

```
int[] myIntArray = new int[5];
String[] myStringArray = new String[5];

myIntArray[0] = 2;
myIntArray[1] = 3;

myStringArray[0] = "Hello";
myStringArray[1] = "World";
```

Declaring and Assigning values in one statement:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

2D and 3D arrays also follow the same principle for declarations.

2D array

```
<datatype>[][] <array_identifier> = new datatype[row][col];
```

3D array, can you name the daily usage of such array?

```
<datatype>[][][] <array_identifier> = new datatype[channel][row][col];
```

Task:

(on Loop)

1. Write a program that takes an integer (say, n) and your name as inputs, and then prints your name n times to the console.

2. Write a program that prints the following patterns:

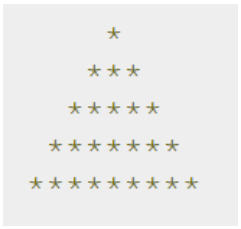
(a)

```
5 4 3 2 1
  4 3 2 1
    3 2 1
      2 1
        1
```

(b)

```
1
  1 2
  1 2 3
  1 2 3 4
  1 2 3 4 5
```

(c)



(d)

3. Write a program that takes an integer and prints its divisors, i.e. divisors of 12 are 1, 2, 3, 4, 6.

4. Write a program which will use while loop to print all the integers between 100 and 150 which
are divisible by 8 in descending order.
Output: 144 136 128 120 112 104

(on Array)

1. Declare an integer array of size 6, initialize it with user input, calculate and print the average. Now calculate the percentage of numbers that are above that average.

For example: if 3 of the array elements are greater than average, percentage is: 3 * 100 / 6 = 50%

2. Take an integer from user, generate that many Fibonacci numbers and store in an array. Display the array.

Sample output:

```
Enter a number: 8
First 8 Fibonacci numbers: 0 1 1 2 3 5 8 13
```

3. Take a 3X3 array and initialize it with these values:

3 4 9
4 6 0
2 9 11

Calculate and print the sum for each row, column and both diagonals.

4. Take an integer array and print only the numbers that are in consecutive orders of 3.

```
Enter size: 12
Enter numbers: 1 2 3 2 2 2 11 4 4 4 3 3
Output: 2 4
```