**Tutorials for Chapters 5 & 6**

1. Write a method with the name "**shape\_print**” that will get an integer argument (n), then it should print an (nXn) characters of a box (row X column) as a sample output given below. Your method should not return any value to back. (8 points).

**Rule**: In that box, only diagonal and reverse diagonal characters will be ‘X’, the remaining characters should be blank character (“ “).

 **Example**: (if n is given as 8)

 shape\_print(8) should give this output:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X |  |  |  |  |  |  | X |
|   | X |  |  |  |  | X |  |
|  |  | X |  |  | X |  |  |
|  |  |  | X | X |  |  |  |
|  |  |  | X | X |  |  |  |
|  |  | X |  |  | X |  |  |
|  | X |  |  |  |  | X |  |
| X |  |  |  |  |  |  | X |

**Solution:**

 public static void shape\_print(int n) { // Header 1 pt

 for(int i=1; i<=n; i++) { // loop-1: 1 pt

 System.out.printf("\n"); // New row starter: 1 pt

 for (int j=1; j<=n; j++) { // loop-2: 1 pt

 if (i==j)||(i+j==(n+1)) System.out.printf("X"); // correct cond. Checking 4pts

 else System.out.printf(" "); // output 2 pts

 }

 }

 System.out.printf("\n");

 }

1. Write a method named “draw\_triangle”. It should get an integer argument n and it should print a triangle with a height as n rows. It should not return any value to back.

There is a the sample run given below for n=5.

**Sample Run:**

draw\_triangle(5)

 \*

 \*\*

 \*\*\*

 \*\*\*\*

\*\*\*\*\*

**Solution:**

**Observations:**

1. We will use two loops with counters i and j
2. When j<(n-i+1) it should print blank character (‘ ’) else it should print ‘\*’ character

 public static void box\_print(int n) {

 for(int i=1; i<=n; i++) {

 System.out.printf("\n");

 for (int j=1; j<=n; j++) {

 if (j<n-i+1) System.out.printf(" ");

 else System.out.printf("\*");

 }

 }

 }

1. Write a method named “draw\_squire”. It should get an integer argument n and it should print a squire with height=n, width=n. It should not return any value to back.

There is a the sample run given below for n=5.

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Observations:**

1. We will use two loops with counters i and j
2. When j==1 or j==n or i==1 or i==n print ‘\*’ else print blank character (‘ ’)

 public static void box\_print(int n) {

 for(int i=1; i<=n; i++) {

 System.out.printf("\n");

 for (int j=1; j<=n; j++) {

 if (j==1 || j==n || i==1 || i==n)

 System.out.printf("\*");

 else

 System.out.printf(" ");

 }

 }

 }

1. Write a method named “list\_odd\_numbers”. It should get two integer argument min, max and it should found and print all odd numbers between min and max (including min and max). It should not return any value to back.

There is a sample run given below for min=12, and max=22.

13, 15, 17, 19, 21

**Solution:**

**Observations:**

1. We will one loop with counter “cnt” that will start from min and ended at max.
2. When (cnt%2==1) it is an odd number so print it!

 public static void list\_odd\_numbers(int min, int max) {

 for(int cnt=min; cnt<=max; cnt++) {

 if (cnt%2==1)

 System.out.printf("%3d, "cnt);

 }

 }

1. Write a method with a name “factorial”. It should get an Integer argument named "num".

It should calculate the num!= num \* (num-1) \* (num-2) \* ... \* 2 \* 1

Finally your method should return the calculated value to back.

**Solution:**

 public static int factorial(int num) {

 int i;

 int factor=1;

 for (i=2;i<=num;i++) {

 factor\*=i; }

 return factor;

 }

1. Write a method “**game**” that will get in no argument but will return the score of the turn (returns 0 when equality, 1 when computer wins, -1 when user wins). Your method will play a single turn of Paper-Scissor-Rock game. (10 points).
* First, it should ask user her selection to enter (0 for Paper, 1 for Scissor and 2 for Rock).
* Then it should generate a random integer 0,1 or 2 for computer selection (0 for Paper, 1 for Scissor and 2 for Rock).
* Then it should decide who wins the turn based on the rules given below and return 0 on equality, 1 when computer wins, -1 when user wins 0.

The game rules:

* Paper(0) covers the rock(2), so paper(0) wins.
* Scissor(1) cuts the paper(0), so scissor(1) wins
* Rock(2) breaks the scissor(1), so rock(2) wins

 **Solution:**

public static int game() {

 int max=2, min=0, user\_ch, comp\_ch, tot;

 String str;

 str = JOptionPane.showInputDialog(null,

 "Enter your choice (0:Paper, 1:Scissor and 2:Paper)");

 user\_ch=Integer.parseInt(str);

 comp\_ch =

 (int) Math.floor(Math.random()\*(max-min+1)+ min);

 System.out.printf("(0:Paper, 1:Scissor and 2:Paper)"+

 "Computer Choice =%d,"+

 "User Choice=%d", comp\_ch, user\_ch);

 tot = comp\_ch-user\_ch;

 if (tot==0) return 0; //equality

 if (tot==1) return 1; // computer wins

 if (tot==-1) return -1; // user wins

 if (tot==2) return -1; // user wins ( Rock-Paper condition)

 if (tot==-2) return 1; // computer wins (Paper – Rock condition)

 return 0;

}

TRY to do the if (...) part in another (longer) way as a SELF WORK !!!