

LAB 6

Java Methods

Name	:	Lee Pei Hui
Matric Number	:	22004781/1
Occurrence	:	7
Lecturer	:	Dr. Soraya

1. Write a Java method that returns a triangular number. A triangular number is defined as $1+2+3+\dots+n$. Then, write a Java program to use the method to display the first 20 triangular numbers.

```
public class L6Q1 {  
    public static void main(String[] args){  
  
        L6Q1 calc = new L6Q1();  
  
        for(int i = 0; i<=20; i++){  
            System.out.println(calc.triangularNumber(i));  
        }  
    }  
  
    int triangularNumber(int n){  
        return (n*(n+1))/2;  
    }  
}
```

2. Write a Java method `multiPrint(int n, char c)` that prints `n` copies of character `c`. Then, write a Java program to use the method to display the triangles and diamonds.

```
public class L6Q2 {  
    public static void main(String[] args){  
  
        L6Q2 shape = new L6Q2();  
  
        for(int i=0; i<5; i++) {  
            shape.multiPrint((i+1),'*');  
        }  
  
        System.out.println(" * ");  
        System.out.println(" *** ");  
        shape.multiPrint(5,'*');  
        System.out.println(" *** ");  
        System.out.println(" * ");  
  
    }  
  
    void multiPrint(int n, char c) {  
        System.out.println(String.valueOf(c).repeat(n));  
    }  
}
```

3. Write a Java method that accepts an array of 10 integers. The method should reverse the integer in the array. Example, if the number is 1234, the number will change to 4321.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Please enter a string of numbers: ");
        String num = in.nextLine();
        reverseStr(num);
    }

    static void reverseStr(String str){
        char[] arr = str.toCharArray();
        for(int i = arr.length-1; i>=0; i--){
            System.out.print(arr[i]);
        }
    }
}
```

4. Write a Java method that implements Euclidean Algorithm to return the greatest common divisor of two positive integers. Then, write a program to get the GCD for **(24, 8) and (200, 625)**.

```
public class L6Q4{  
    public static void main(String[] args){  
        System.out.println(euclid(24,8));  
        System.out.println(euclid(200,625));  
    }  
    public static int euclid(int x, int y) {  
        if (x == 0 || y == 0) {  
            return 1;  
        }  
  
        if (x < y) {  
            int t = x;  
            x = y;  
            y = t;  
        }  
  
        if (x % y == 0) {  
            return y;  
        } else {  
            return euclid(y, x % y);  
        }  
    }  
}
```

5. Write a Java method that accepts three parameters, the method will compare whether the third parameter value is equal to the multiplication of parameter 1 and parameter 2. Then, write a Java multiplication game for any random number within 0 – 12.

Example Output:

```
Enter negative number to quit.  
5 x 8 = 40  
Enter negative number to quit.  
7 x 9 = 16  
Enter negative number to quit.  
6 x 6 = 36  
Enter negative number to quit.  
3 x 2 = -1  
Your Score is 2
```

```
//Java Sample Output  
import java.util.Scanner;  
  
public class L6Q5{  
    static int score = 0;  
  
    public static void main(String[] args){  
        Scanner in = new Scanner(System.in);  
        int a,b,c;  
  
        do {  
            System.out.println("\nEnter negative number to quit. ");  
            a = in.nextInt();  
            b = in.nextInt();  
            c = in.nextInt();  
  
            multiplication(a,b,c);  
  
            if (a < 0 || b < 0 || c < 0) {  
                multiplication(a,b,c);  
                break;  
            }  
  
        }while(a > 0 && b > 0 && c > 0);  
  
        System.out.println("\nYour Score is "+score);  
    }  
  
    public static void multiplication(int a, int b, int c){  
        if(a * b == c){  
            score++;  
        }  
        System.out.printf("%d x %d = %d",a,b,c);  
    }  
}
```

```
//Java Multiplication game
public class L6Q5{
    static int score = 0;

    public static void main(String[] args){
        int a,b,c;

        System.out.println("\nEnter negative number to quit. ");
        a = (int)(Math.random()*12)+1;
        b = (int)(Math.random()*12)+1;
        c = (int)(Math.random()*12)+1;

        multiplication(a,b,c);

        System.out.println("\nYour Score is "+score);
    }

    public static void multiplication(int a, int b, int c){
        if(a * b == c){
            score++;
        }
        System.out.printf("%d x %d = %d",a,b,c);
    }
}
```

6. Write a Java method that determine whether a number is a palindromic prime and another method that determine whether a number is emirp (the number is a prime number and the reverse also a prime number and is not palindromic prime). Then, write a Java program to use the methods to display the first 20 palindromic prime and emirp.

```
public class L6Q6{
    public static void main(String[] args){
        L6Q6 numbers = new L6Q6(); //Create object

        numbers.displayPalindromePrime(20); //Display 20 Palindrome Prime
        numbers.displayEmirp(20); //Display 20 Emirp
    }

    //PALINDROME PRIME
    void displayPalindromePrime(int n){
        int i = 0, thisNumber=2;
        System.out.println("First 20 Palindrome Primes:");
        while(i<n){
            if((((int)Math.log10(thisNumber)+1)%2==0) && !(((int)Math.log10(thisNumber)+1)==2)){
                thisNumber = (int) Math.pow(10, (int)Math.log10(thisNumber)+1);
            }
            if ((thisNumber == reverse(thisNumber)) && isPrime(thisNumber)) {
                System.out.printf("%d%s", thisNumber, (i < (n - 1)) ? ", " : "\n");
                i++;
            }
            thisNumber++;
        }
    }

    //EMIRP
    void displayEmirp(int n){
        int i = 0, thisNumber = 13;
        System.out.println("First 20 Emirps: ");
        while(i<n){
            if(isPrime(thisNumber) && thisNumber != reverse(thisNumber) && isPrime(reverse(thisNumber))){
                System.out.printf("%d%s", thisNumber, (i<(n-1)) ? ", ":"\n");
                i++;
            }

            thisNumber++;
        }
    }
}
```



```
//REVERSE
static int reverse(int n){
    int rev = 0;
    while(n>0){
        rev = rev * 10 + n % 10;
        n /= 10;
    }
    return rev;
}

static boolean isPrime(int n){
    for(int i=2; i<Math.sqrt(n); i++){
        if(n%i == 0){ return false;}
    }
    return true;
}
}
```

Link:

<https://github.com/PeiHui369/Fundamentals-Of-Programming/tree/main/Lab%206>