OOP LAB

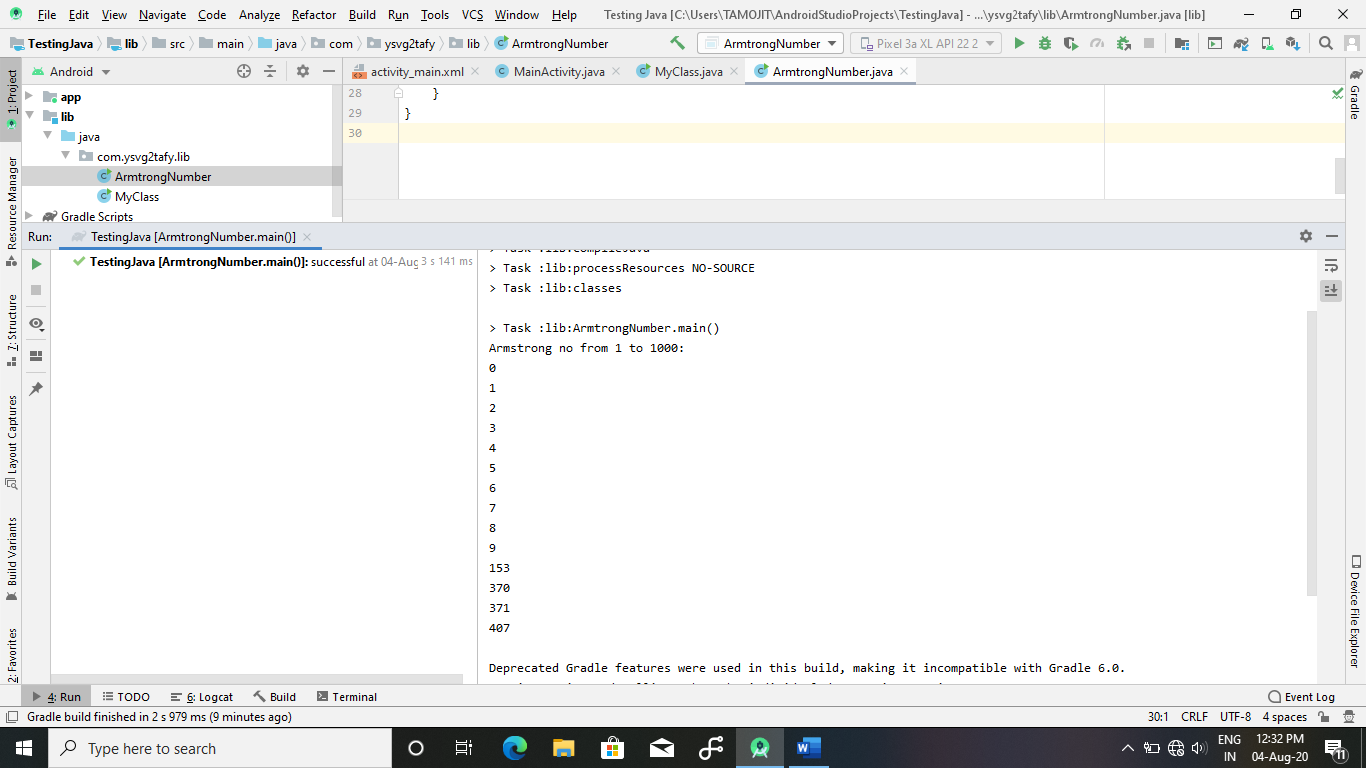
Tamojit Das ; Sec: A ; Roll: 7; Grp: C1

Assignment 1

1. Armstrong Number

Code:

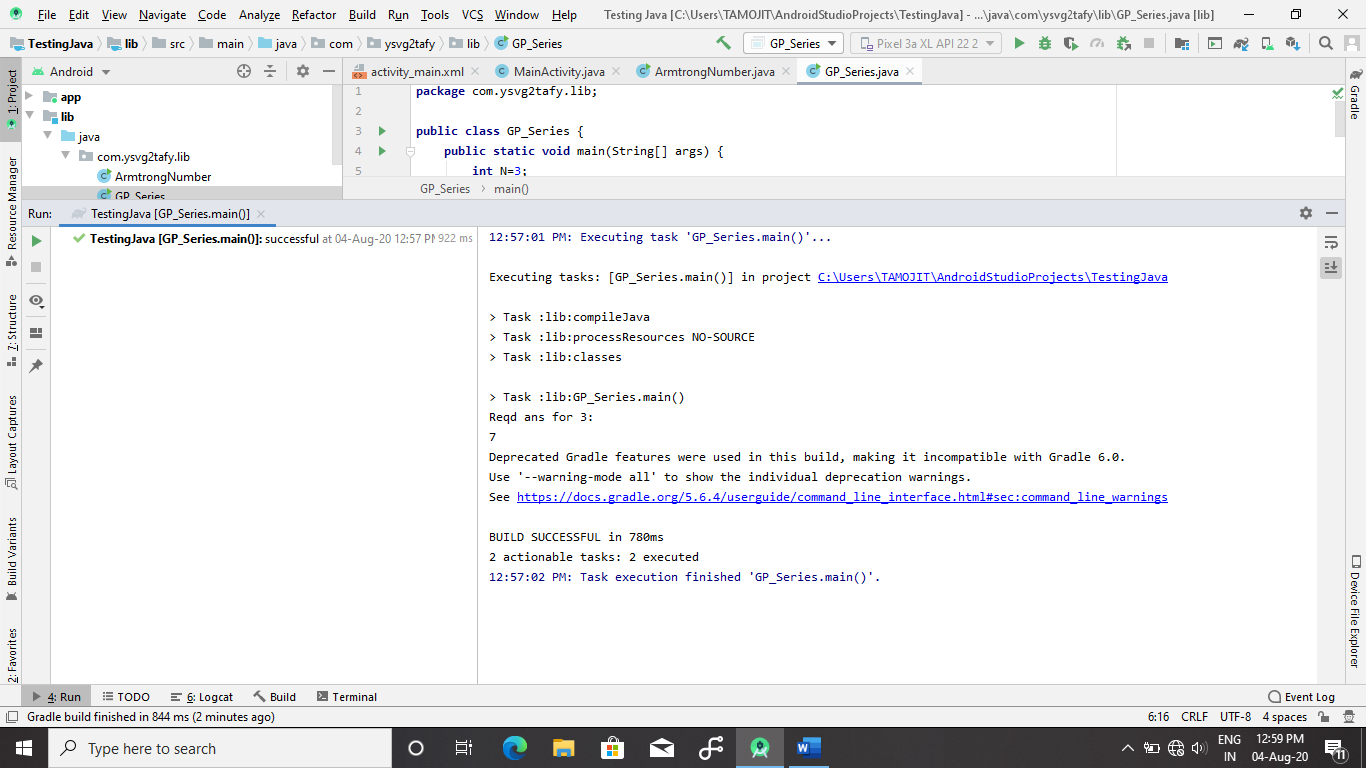
**public class** ArmtrongNumber {  
  
 **public static void** main(String[] args) {  
 System.***out***.println(**"Armstrong no from 1 to 1000:"**);  
 **for** (**int** i=1;i<=1000;i++){  
 **if**(*checkArm*(i)){  
 System.***out***.println(i);  
 }  
 }  
 }  
  
 **private static boolean** checkArm(**int** i) {  
 **int** length=0;  
 **int** copy=i;  
 **while**(copy>0){  
 length++;  
 copy=copy/10;  
 }  
 copy=i;  
 **int** sum=0;  
 **while** (copy>0){  
 sum= (**int**) (sum+Math.*pow*((copy%10),length));  
 copy=copy/10;  
 }  
 **return** sum == i;  
 }  
}



1. GP series: 1+x2+x3+…

Source Code:

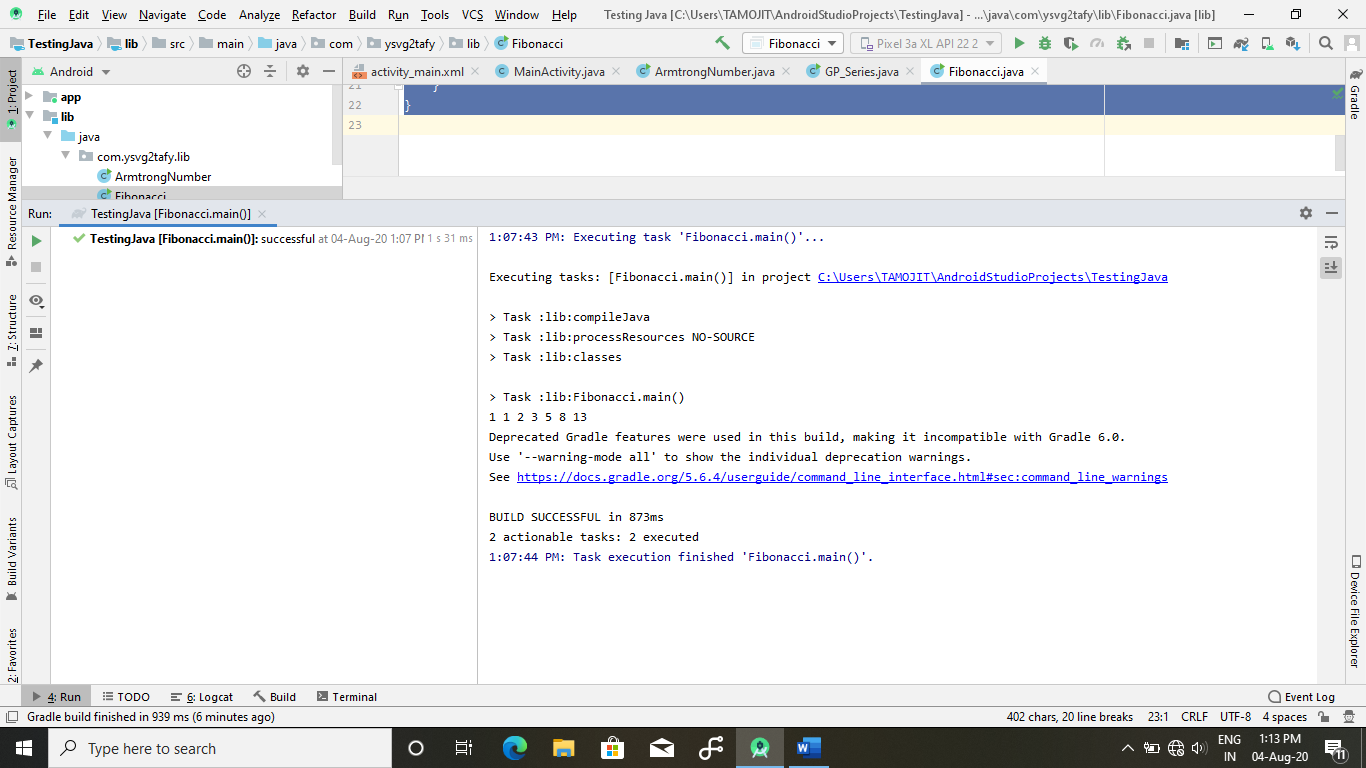
**public class** GP\_Series {  
 **public static void** main(String[] args) {  
 **int** N=3;  
 **int** x=2;  
 **int** ans=*compute*(N,x);  
 *println*(**"Reqd ans for "**+N+**":"**);  
 *print*(ans);  
 }  
  
 **private static int** compute(**int** N, **int** x) {  
 **return** (**int**) ((Math.*pow*(x,N)-1)/(x-1));  
 }  
  
 **private static void** print(**int** x) {  
 System.***out***.print(x);  
 }  
  
 **private static void** println(String x) {  
 System.***out***.println(x);  
 }  
}



1. Fibinacci

Source Code:

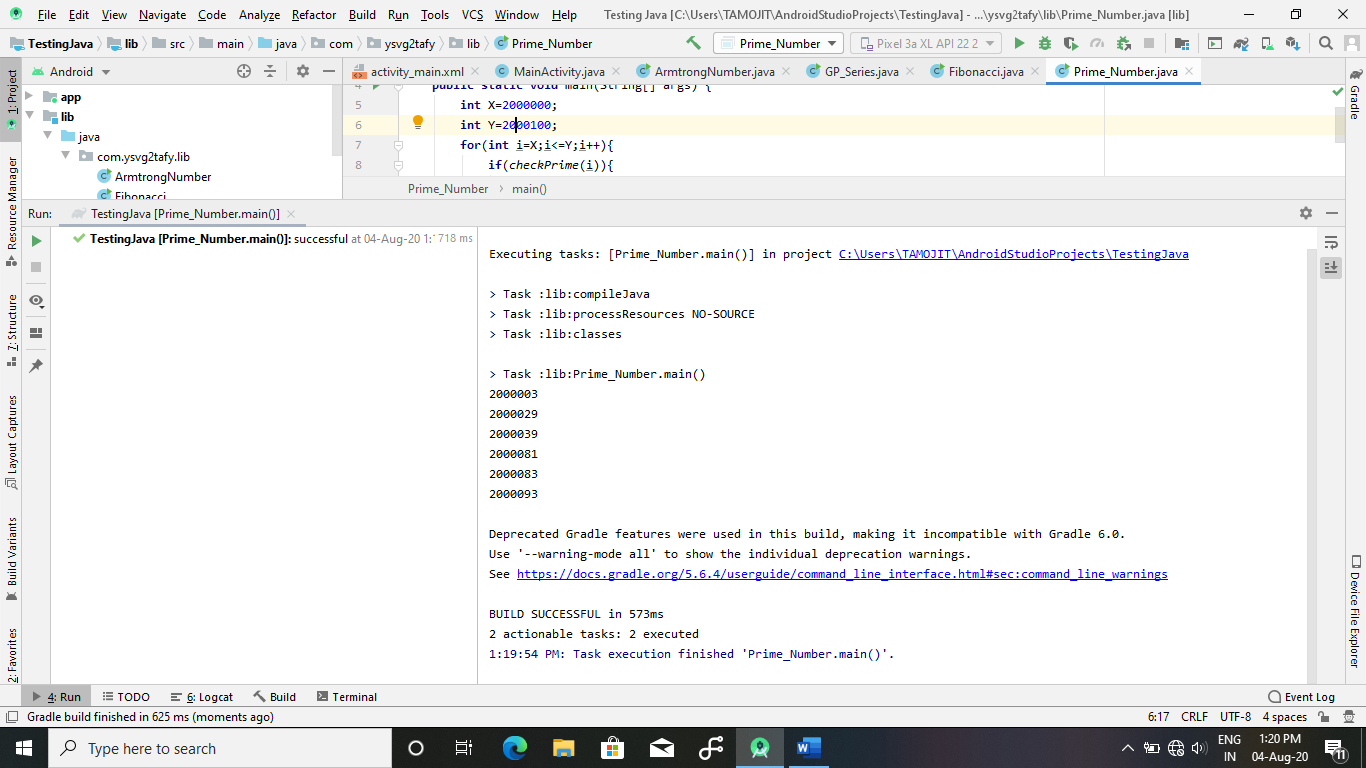
**public class** Fibonacci {  
 **public static void** main(String[] args) {  
 **int** N=7;  
 *showFibi*(N);  
 }  
  
 **private static void** showFibi(**int** n) {  
 **for**(**int** i=1;i<=n;i++){  
 System.***out***.print(*Fib*(i)+**" "**);  
 }  
 }  
  
 **private static int** Fib(**int** i) {  
 **if**(i==1 || i==2){  
 **return** 1;  
 }**else**{  
 **return** *Fib*(i-1)+*Fib*(i-2);  
 }  
 }  
}



1. Prime Numbers

Source Code:

**public class** Prime\_Number {  
 **public static void** main(String[] args) {  
 **int** X=2000000;  
 **int** Y=2000100;  
 **for**(**int** i=X;i<=Y;i++){  
 **if**(*checkPrime*(i)){  
 System.***out***.println(i);  
 }  
 }  
 }  
  
 **private static boolean** checkPrime(**int** x) {  
 **for**(**int** i=2;i<(**int**)Math.*pow*(x,0.5)+1;i++){  
 **if**(x%i==0){  
 **return false**;  
 }  
 }  
 **return true**;  
 }  
}



1. Gcd

Source Code

**public class** GCD {  
 **public static void** main(String[] args) {  
 **int** X=36488521;  
 **int** Y=37852256;  
 System.***out***.println(*computeGCD*(X,Y));  
 }  
  
 **private static int** computeGCD(**int** x, **int** y) {  
 **if**(x%y==0){  
 **return** y;  
 }**else**{  
 **return** *computeGCD*(y,x%y);  
 }  
 }  
}

