RIGHTSTROKE

**Assignment 1**

1.Write a java program to find the area of rectangle

Program:

import java.util.Scanner;

public class Program

{

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int width=sc.nextInt();

int height=sc.nextInt();

int area=width\*height;

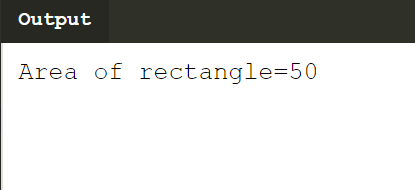
System.out.println("Area of rectangle="+area);

}

}

Output:

Inputs: 5 10



2. Write a java program to check the given no is Armstrong or not(153 is Armstrong no

1\*1\*1+5\*5\*5+3\*3\*3=153)

Program:

import java.util.Scanner;

class Armstrong{

public static void main(String[] args) {

int s=0,r,temp;

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

temp=n;

while(n>0)

{

r=n%10;

s=s+(r\*r\*r);

n=n/10;

}

if(temp==s)

System.out.println("armstrong number");

else

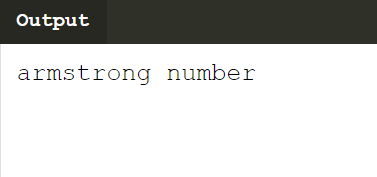
System.out.println("Not armstrong number");

}

}

Output:

Input: 153



3. Write a java program to check the given no is palindrome or not

Program:

import java.util.Scanner;

class Palindrome{

public static void main(String[] args) {

int s=0,r,temp;

Scanner sc= new Scanner(System.in);

int n=sc.nextInt();

temp=n;

while(n>0){

r=n%10;

s=(s\*10)+r;

n=n/10;

}

if(temp==s)

System.out.println("palindrome number ");

else

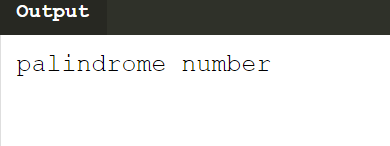
System.out.println("not palindrome");

}

}

Output:

Input: 121



4.Write a java program to generate first N prime numbers

Program: import java.util.Scanner;

public class Program

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

if(n==1)

System.out.print(2);

else

{

System.out.print(2+" ");

int c=1,i,j,x=3;

while(true)

{

if(c==n)

break;

j=1;

for(i=2;i<=x/2;i++)

{

if(x%i==0)

{

j=0;

break;

}

}

if(j==1)

{

System.out.print(x+" ");

c+=1;

}

x+=1;

}

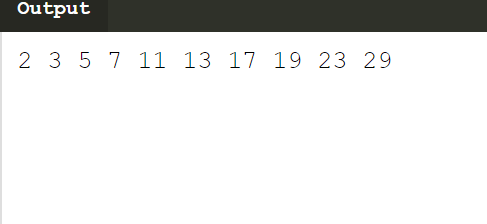
}

}

}

Output:

Intput:10



5.Write a java program to print even numbers in between given two numbers.

Program:

import java.util.Scanner;

public class EvenNumbers {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int start=sc.nextInt();

int end=sc.nextInt();

int i;

System.out.println("Printing Even numbers between given two number "+start+"and " + end);

for(i=start; i<= end; i++){

if(i % 2 == 0){

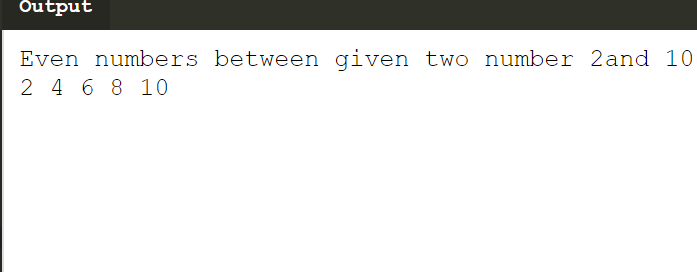
System.out.print(i + " ");

}

}

}

}

Output:  
 Input: 2 10 

**Theory**

1.What is Abstraction?

Answer: Abstraction is the concept of hiding the internal details and describing things in simple

terms. For example, a method that adds two integers. The internal processing of the

method is hidden from the outer world.

2. What is Encapsulation?

Answer: Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting

on the data (methods) together as a single unit.

To achieve encapsulation in Java –

* + - Declare the variables of a class as private.
    - Provide public setter and getter methods to modify and view the variables

values.

3. What is JDK?

Answer: The Java Development Kit (JDK) is a software development environment used for

developing Java applications and applets. It includes the Java Runtime Environment

(JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation

generator (javadoc) and other tools needed in Java development.

4. What is JVM?

Answer: JVM (Java Virtual Machine) is an abstract machine that enables your computer to run a

Java program.

When you run the Java program, Java compiler first compiles your Java code to

bytecode. Then, the JVM translates bytecode into native machine code (set of

instructions that a computer's CPU executes directly).

5. Define Inheritance?

Answer: **Inheritance in Java** is a mechanism in which one object acquires all the properties

and behaviors of a parent object

The idea behind inheritance in Java is that you can create new [classes](https://www.javatpoint.com/object-and-class-in-java) that are built

upon existing classes. When you inherit from an existing class, you can reuse methods

and fields of the parent class. Moreover, you can add new methods and fields in your

current class also.

6. How java achieved platform independence?

Answer: Since every Java program runs on Java virtual machine, same byte code can be run on

any platform. key is byte code is not machine instruction they are platform

independent instruction to JVM  In summary combination of byte code and JVM

makes Java program platform independent

7. Write the syntax of main function.

Answer: public static void main(final String[] args)

Java main method is the entry point of any java program. Its syntax is always public

static void main(String[] args). You can only change the name of String array argument.

for example you can change args to myarg.

8. What is conditional operator?

Answer: The conditional operator is also known as the ternary operator. This operator consists of

three operands and is used to evaluate Boolean expressions. The goal of the operator is

to decide; which value should be assigned to the variable. The operator is written as:

variable x = (expression)? value if true: value if false

9. How many data types in java?

Answer: In Java, we have eight primitive data types: boolean, char, byte, short, int, long, float

and double. Java developers included these data types to maintain the portability of

java as the size of these primitive data types do not change from one operating system

to another.

**byte**, **short**, **int** and **long** data types are used for storing whole numbers.

**float** and **double** are used for fractional numbers.

**char** is used for storing characters(letters).

**boolean** data type is used for variables that holds either true or false.

10. What is constant? How it is declared?

Answer: Constants in Java are used when a **‘**[**static**](https://www.edureka.co/blog/variables-in-java/)**‘** value or a permanent value for a variable has

to be implemented. Java doesn’t directly support constants. To make any variable a

constant, we must use ‘static’ and ‘final’ modifiers in the following manner:

**Syntax to assign a constant value in java:**

static final datatype identifier\_name = constant;

* The **static modifier** causes the variable to be available without an instance of it’s

defining class being loaded

* The **final modifier** makes the variable unchangeable