1) **Write a java program to find area of the rectangle**?

import java.util.Scanner;

class AreaOfRectangle {

public static void main (String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter length of Rectangle:");

double length = sc.nextDouble();

System.out.println("Enter width of Rectangle:");

double width = sc.nextDouble();

double area = length\*width;

System.out.println("Area of Rectangle is:"+area);

}

}

**OUTPUT**: -

Enter length of Rectangle:

2

Enter width of Rectangle:

8

Area of Rectangle is:16.0

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)**

public class JavaExample {

public static void main(String[] args) {

int num = 153, number, temp, total = 0;

number = num;

while (number != 0)

{

temp = number % 10;

total = total + temp\*temp\*temp;

number /= 10;

}

if(total == num)

System.out.println(num + " is an Armstrong number");

else

System.out.println(num + " is not an Armstrong number");

}

}

**OUTPUT:** -

153 is an Amstrong number

3)**Write a java program to check the given no is palindrome or not?**

public class Palindrome {

public static void main(String[] args) {

int num = 121, reversedInteger = 0, remainder, originalInteger;

originalInteger = num;

while( num != 0 )

{

remainder = num % 10;

reversedInteger = reversedInteger \* 10 + remainder;

num /= 10;

}

if (originalInteger == reversedInteger)

System.out.println(originalInteger + " is a palindrome.");

else

System.out.println(originalInteger + " is not a palindrome.");

}

}

**OUTPUT**: -

1. a palindrome number.

4) **Write a java program to generate first N prime numbers?**

import java.util.Scanner;

class PrimeNumberDemo

{

public static void main(String args[])

{

int n;

int status = 1;

int num = 3;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the value of n:");

n = scanner.nextInt();

if (n >= 1)

{

System.out.println("First "+n+" prime numbers are:");

//2 is a known prime number

System.out.println(2);

}

for ( int i = 2 ; i <=n ; )

{

for ( int j = 2 ; j <= Math.sqrt(num) ; j++ )

{

if ( num%j == 0 )

{

status = 0;

break;

}

}

if ( status != 0 )

{

System.out.println(num);

i++;

}

status = 1;

num++;

}

}

}

**OUTPUT**: -

Enter the value of n:

15

First 15 prime numbers are:

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

5) **Write a java program to print even numbers in between given two numbers?**

class JavaExample {

public static void main(String args[]) {

int n = 100;

System.out.print("Even Numbers from 1 to "+n+" are: ");

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

if (i % 2 == 0) {

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

}

}

}

}

**OUTPUT**: -

Even Numbers from 1 to 100 are: 2 4 6 8 10 12 14 16 18 20 22 24 26 28

30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76

78 80 82 84 86 88 90 92 94 96 98 100

1**) ABSTRACTION**: -

Abstraction is one of the key concepts of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. In an OOP language provide an abstraction that hides the internal implementation details.

2) **ENCAPSULATION**: -

Encapsulationin java is a *process of wrapping code and data together into a single unit*, for example, a capsule which is mixed of several medicines. We can create a fully encapsulated class in Java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

3) **JDK:** -

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle corporation:

* Standard Edition Java Platform
* Enterprise Edition Java Platform
* Micro Edition Java Platform

4) **JVM**: -

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

JVMs are available for many hardware and software platforms. There are three notions of the JVM: *specification*, *implementation*, and *instance*.

The JVM performs the following main tasks:

* Loads code
* Verifies code
* Executes code
* Provides runtime environment

5) **INHERITANCE**: -

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes 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.

Inheritance represents the *parent-child* relationship.

**Terms used in Inheritance**: -

**Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.

**Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.

6) **How java achieved platform independence**?

When you write program in JAVA and when you compile it, a separate file is created for the program compiled, this file is called .class file is known as byte code of java. This byte code will not be in executable stage. The main purpose of generating byte code for a program compiled is to achieve platform independency that means this byte code generated in one platform can be executed in another. The one which makes the byte code generated in Windows OS to be executed in the UNIX OS is the JVM of UNIX platform. From this statement, you may have understood that JVM is platform dependent and the byte code generated by Java program is platform independent. The Byte code generated can run on any JVM irrespective of to which platform the JVM belongs.

7) **Write the syntax of the main function**?

public static void main (String args []);

**public:** It is an access specifier. We should use a public keyword before the main() method so that JVM can identify the execution point of the program. If we use private, protected, and default before the main() method, it will not be visible to JVM.

**static:** You can make a method static by using the keyword static. We should call the main() method without creating an object. Static methods are the method which invokes without creating the objects, so we do not need any object to call the main() method.

**void:** In Java, every method has the return type. Void keyword acknowledges the compiler that main() method does not return any value.

**main():** It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

**String args []:** The main() method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

8) **What is conditional operator**?

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 datatypes are there in java**?

In java there are 8 primitive data types and 3 non-primitive datatypes are present

Primitive Datatypes are Boolean, Char, Byte, Short, Int, Long, Float, Double.

Non-Primitive Datatypes are String, Array, User defined classes.

10) **What is constant and how it is declared**?

Constant is a fixed value that cannot be changed. One example of a constant is pi, because it always has the same value, which is 3.14159… This concept of constants is relevant to us, because we often need to declare constants when we write programs in Java.

**How to declare constant: -**

1. **To turn an ordinary variable into a constant, you have to use the keyword "final."**
2. **As a rule, we write constants in capital letters to differentiate them from ordinary variables.**
3. **If you try to change the constant in the program, the Java Compiler sends an error message. This happens because you can only assign a value to a constant once.**