*Q1. What are different buzz words of Java? Explain ?*

*The Java programming language is a high-level language that can be characterized by all of the following buzzwords:*

*1. Simple*

* *Java was designed to be easy for a professional programmer to learn and use effectively.*
* *It’s simple and easy to learn if you already know the basic concepts of Object Oriented Programming.*
* *Best of all, if you are an experienced C++ programmer, moving to Java will require very little effort. Because Java inherits the C/C++ syntax and many of the object-oriented features of C++, most programmers have little trouble learning Java.*
* *Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.*

*2. Object Oriented*

* *Java is true object-oriented language.*
* *Almost “Everything is an Object” paradigm. All program code and data reside within objects and classes.*
* *The object model in Java is simple and easy to extend.*
* *Java comes with an extensive set of classes, arranged in packages that can be used in our programs through inheritance.*
* *Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.*

*Basic concepts of OOPs are:*

* *Object*
* *Class*
* *Inheritance*
* *Polymorphism*
* *Abstraction*
* *Encapsulation*

*3. Distributed*

* *Java is designed fa or distributed environment of the Internet. Its used for creating applications on networks.*
* *Java applications can access remote objects on the Internet as easily as they can do in the local system.*
* *Java enables multiple programmers at multiple remote locations to collaborate and work together on a single project.*
* *Java is designed for the distributed environment of the Internet because it handles TCP/IP protocols.*

*4. Compiled and Interpreted*

* *Usually, a computer language is either compiled or Interpreted. Java combines both this approach and makes it a two-stage system.*
* *Compiled: Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java Bytecode.*
* *Interpreted: Bytecode is then interpreted, which generates machine code that can be directly executed by the machine that provides a Java Virtual machine.*

*5. Robust*

* *It provides many features that make the program execute reliably in a variety of environments.*
* *Java is a strictly typed language. It checks code both at compile time and runtime.*
* *Java takes care of all memory management problems with garbage collection.*
* *Java, with the help of an exception handling, captures all types of serious errors and eliminates any risk of crashing the system.*

*6. Secure*

* *Java provides a “firewall” between a networked application and your computer.*
* *When a Java Compatible Web browser is used, downloading can be done safely without fear of viral infection or malicious intent.*
* *Java achieves this protection by confining a Java program to the Java execution environment and not allowing it to access other parts of the computer.*

*7. Architecture Neutral*

* *Java language and Java Virtual Machine helped in achieving the goal of “write once; run anywhere, any time, forever.”*
* *Changes and upgrades in operating systems, processors and system resources will not force any changes in Java Programs.*

*8. Portable*

* *Java Provides a way to download programs dynamically to all the various types of platforms connected to the Internet.*
* *It helps in generating Portable executable code.*

*9. High Performance*

* *Java performance is high because of the use of bytecode.*
* *The bytecode was used so that it was easily translated into native machine code.*

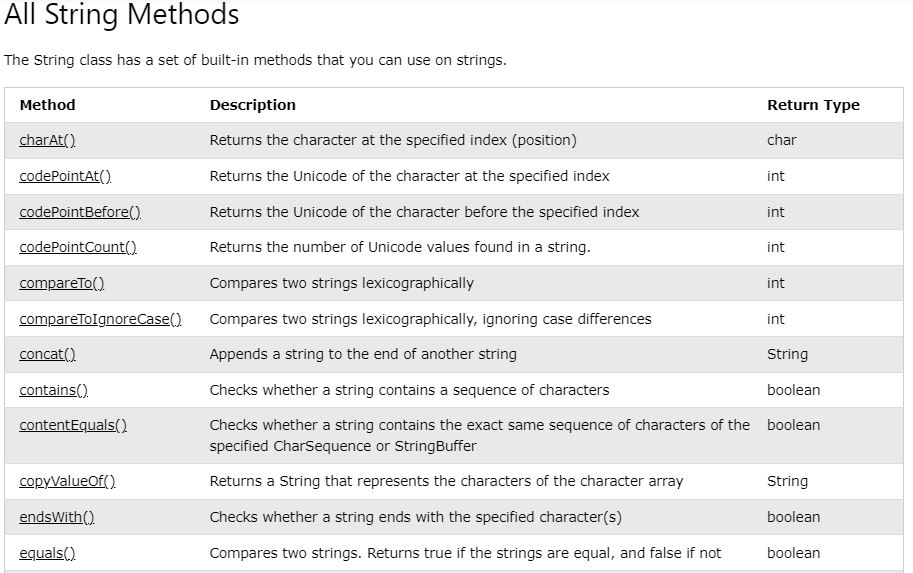
*10. Multithreaded*

* *Multithreaded Programs handled multiple tasks simultaneously, which was helpful in creating interactive, networked programs.*
* *Java run-time system comes with tools that support multiprocess synchronization used to construct smoothly interactive systems.*

*11. Dynamic*

* *Java is capable of linking in new class libraries, methods, and objects*

*Q2)*  ***what******is string class? Explain different methods of string class****.*

**

*Q3. explain Java virtual machine?*

*JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.*

*JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).*

*A specification where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.*

*An implementation Its implementation is known as JRE (Java Runtime Environment).*

*Runtime Instance Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.*

*Q4.write Java program which demonstrate use of nextInt, nextFloat and nextLine method.*

*Q5. write a Java program to display following output ?*

*1*

*1 2*

*1 2 3*

*1 2 3 4*

*public class Main {*

*public static void main(String[] args) {*

*int rows = 5;*

*for (int i = 1; i <= rows; ++i) {*

*for (int j = 1; j <= i; ++j) {*

*System.out.print(j + " ");*

*}*

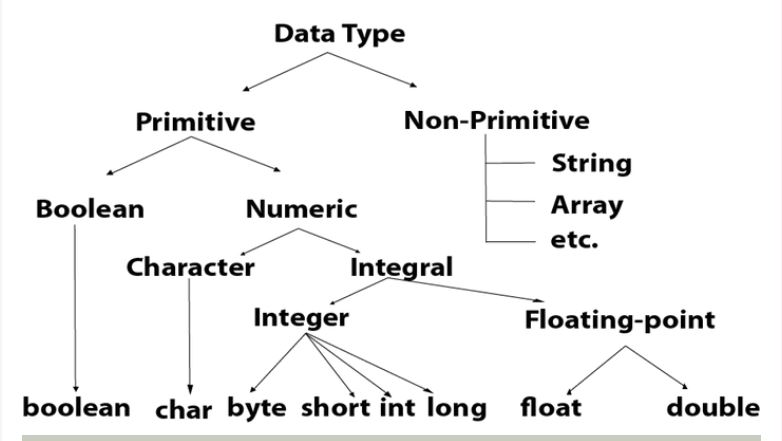
*System.out.println();*

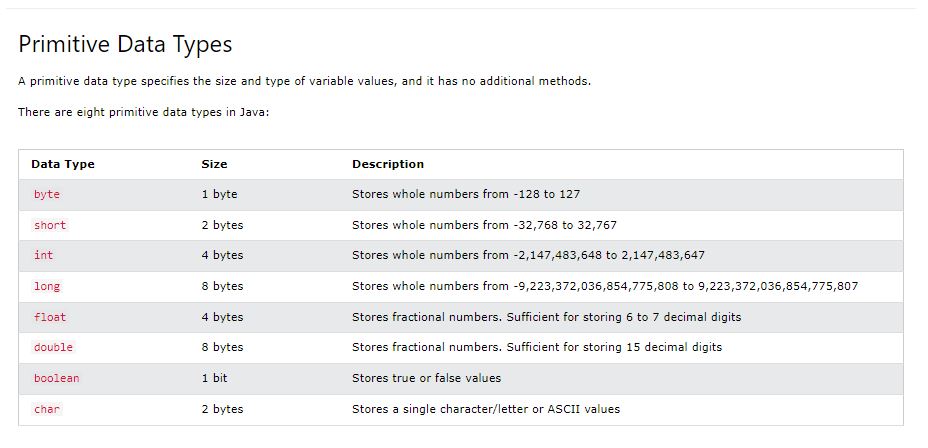
*}*

*}*

*}*

*Q6) Explain data types in Java*

**

**

*Q7) Explain how Java is portable language.*

*Java is portable,i.e. platform independent due to following reasons:*

*1)Output of a Java compiler is bytecode which is non executable code.*

*2) Secondaly,Bytecode is a highly optimized set of instructions*

*3)Bytecode is executed by Java run-time system, which is called the Java Virtual Machine (JVM).*

*4)JVM is an interpreter.*

*5)JVM accepts Bytecode as input and execute it.*

*6)Translating a Java program into bytecode makes it much easier to run a program in a wide variety of environments because only the JVM needs to be implemented for each platform.*

*7)For a given System we have Run-time package , once JVM is installed for particular system then any java program can run on it.*

*Q8) Explain identifiers in Java. or*

*What are the identifiers in Java?*

*Identifiers in Java are names that distinguish between different Java entities, such as classes, methods, variables, and packages. Identifiers include the names of classes, methods, variables, packages, constants, etc. These identifiers are each specified using a specific syntax and naming scheme*

*Q9) Write a Java program to create a object of class and call method area and circumference. Pass necessary parameter to method.*

*import java.util.Scanner;*

*import java.lang.Math;*

*public class AreaOfCircle*

*{*

*public static void main(String[] args)*

*{*

*int option;*

*double radius, circumference, diameter, area;*

*Scanner sc=new Scanner (System.in);*

*System.out.println("1. If the radius is known");*

*System.out.println("2. If the diameter is known");*

*System.out.println("3. If the circumference is known");*

*System.out.print("Enter your choice: ");*

*option=sc.nextInt();*

*switch(option)*

*{ case 1:*

*System.out.print("Enter the radius of the circle: ");*

*radius=sc.nextDouble();*

*area=(Math.PI\*(radius\*radius));*

*System.out.print("The area of the circle is: "+area);*

*break;*

*case 2:*

*System.out.print("Enter the diameter of the circle: ");*

*diameter=sc.nextDouble();*

*area=Math.PI\*(diameter\*diameter)/4;*

*System.out.print("The area of the circle is: "+area);*

*break;*

*case 3:*

*System.out.print("Enter the circumference of the circle: ");*

*circumference=sc.nextDouble();*

*area=(circumference\*circumference)/(4\*Math.PI);*

*System.out.print("The area of the circle is: "+area);*

*break;*

*default:System.out.println("invalid choice!");*

*}*

*}*

*}*

*8)However Internal details of JVM will differ from platform to platform but still all understand the Same Java Bytecode*

*Q10) write a Java program to convert integer to binary number. Use toBinaryString method.*

*import java.util.\*;*

*class Int2Binary {*

*private static String binaryString;*

*private static void int2binary(int n) {*

*binaryString = Integer.toBinaryString(n);*

*}*

*private static void addPadding(){*

*binaryString = String.format("%32s", binaryString).replaceAll(" ", "0");*

*}*

*private static void addBlocks(String separator) {*

*StringBuffer sb = new StringBuffer();*

*for (int i = 0; i< binaryString.length(); i++ ){*

*sb.append(binaryString.charAt(i));*

*if (i%4 == 3 && i != binaryString.length() - 1)*

*sb.append(separator);*

*}*

*binaryString = sb.toString();*

*}*

*public static void convert2Binary(int n) {*

*int2binary(n);*

*addPadding();*

*addBlocks("\_");*

*}*

*public static void main(String[] args) {*

*int n = 10;*

*for (int i = -5; i <= 5; i++){*

*convert2Binary(i);*

*System.out.format("Binary representation of %2d is %s\n", i, binaryString);*

*}*

*}*

*}*

*Q 11) write a program to Demonstrate use of Scanner class in java.*

*Q12) explain bitwise operator with example.*

*// bitwise operators*

*public class operators {*

*public static void main(String[] args)*

*{*

*int a = 5;*

*int b = 7;*

*System.out.println("a&b = " + (a & b));*

*Q13) write a Java program to find largest element in an array.*

*public class LargestElement\_array {*

*public static void main(String[] args) {*

*int [] arr = new int [] {25, 11, 7, 75, 56};*

*int max = arr[0];*

*for (int i = 0; i < arr.length; i++) {*

*if(arr[i] > max)*

*max = arr[i];*

*}*

*System.out.println("Largest element present in given array: " + max);*

*}*

*}*

*Q14) explain instance variable and static variable with example.*

*Java Variables*

*A variable is a container which holds the value while the*[*Java program*](https://www.javatpoint.com/simple-program-of-java)*is executed. A variable is assigned with a data type.*

*Variable is a name of memory location. There are three types of variables in java: local, instance and static.*

*1) Local Variable*

*A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.*

*A local variable cannot be defined with "static" keyword.*

*2) Instance Variable*

*A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as*[*static*](https://www.javatpoint.com/static-keyword-in-java)*.*

*It is called an instance variable because its value is instance-specific and is not shared among instances.*

*3) Static variable*

*A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the*

1. *public class Simple{*
2. *public static void main(String[] args){*
3. *int a=10;*
4. *int b=10;*
5. *int c=a+b;*
6. *System.out.println(c);*
7. *}*
8. *}*

*Q15) What is constructor? Explain constructor overloading in Java with example.*

***Constructor****is a special method that is used to initialize a newly created object and is called just after the memory is allocated for the object. It can be used to initialize the objects to desired values or default values at the time of object creation. It is not mandatory for the coder to write a constructor for a class.*

## *Constructor Overloading in Java*

*Java Constructor overloading is a technique in which a class can have any number of constructors that differ in parameter list*

*class Demo{*

*int value1;*

*int value2;*

*Demo(){*

*value1 = 1;*

*value2 = 2;*

*System.out.println("Inside 1st Parent Constructor");*

*}*

*Demo(int a){*

*value1 = a;*

*System.out.println("Inside 2nd Parent Constructor");*

*}*

*public void display(){*

*System.out.println("Value1 === "+value1);*

*System.out.println("Value2 === "+value2);*

*}*

*public static void main(String args[]){*

*DemoChild d1 = new DemoChild();*

*d1.display();*

*}*

*}*

*class DemoChild extends Demo{*

*int value3;*

*int value4;*

*DemoChild(){*

*//super(5);*

*value3 = 3;*

*value4 = 4;*

*System.out.println("Inside the Constructor of Child");*

*}*

*public void display(){*

*System.out.println("Value1 === "+value1);*

*System.out.println("Value2 === "+value2);*

*System.out.println("Value1 === "+value3);*

*System.out.println("Value2 === "+value4);*

*}*

*}*

*# Arithmetic Operators*

*public class Arithmetic Operators {*

*public static void main(String[] args) {*

*int a = 12, b = 5;*

*System.out.println("a + b = " + (a + b));*

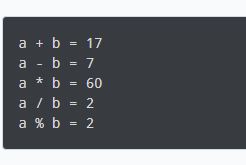
*System.out.println("a - b = " + (a - b));*

*System.out.println("a \* b = " + (a \* b));*

*System.out.println("a / b = " + (a / b));*

*System.out.println("a % b = " + (a % b));*

*}*

*}*

### *# Assignment Operators*

*class Main {*

*public static void main(String[] args) {*

*int a = 4;*

*int var;*

*var = a;*

*System.out.println("var using =: " + var);*

*var += a;*

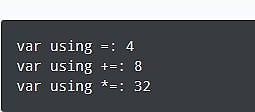
*System.out.println("var using +=: " + var);*

*var \*= a;*

*System.out.println("var using \*=: " + var);*

*}*

*}*

**

*Relational Operators*

*public class ram4 {*

*public static void main(String[] args) {*

*int a = 7, b = 11;*

*System.out.println("a is " + a + " and b is " + b);*

*System.out.println(a == b);*

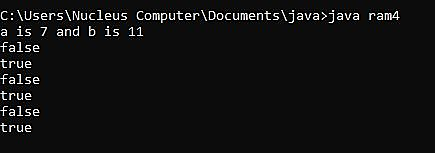
*System.out.println(a != b);*

*System.out.println(a > b);*

*System.out.println(a < b);*

*System.out.println(a >= b);*

*System.out.println(a <= b);*

*}*

*Logical Operators*

*class Main {*

*public static void main(String[] args) {*

*System.out.println((5 > 3) && (8 > 5));*

*System.out.println((5 > 3) && (8 < 5));*

*System.out.println((5 < 3) || (8 > 5));*

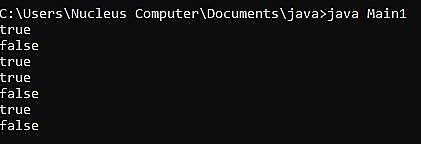
*System.out.println((5 > 3) || (8 < 5));*

*System.out.println((5 < 3) || (8 < 5));*

*System.out.println(!(5 == 3));*

*System.out.println(!(5 > 3));*

*}*

*}*

*Unary Operators*

*class Main3 {*

*public static void main(String[] args) {*

*int a = 12, b = 12;*

*int result1, result2;*

*System.out.println("Value of a: " + a);*

*result1 = ++a;*

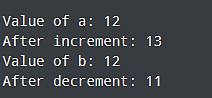
*System.out.println("After increment: " + result1);*

*System.out.println("Value of b: " + b);*

*result2 = --b;*

*System.out.println("After decrement: " + result2);*

*}*

*}*

*#Bit*

*public class operators {*

*public static void main(String[] args)*

*{*

*int a = 5;*

*int b = 7;*

*System.out.println("a&b = " + (a & b));*

*System.out.println("a|b = " + (a | b));*

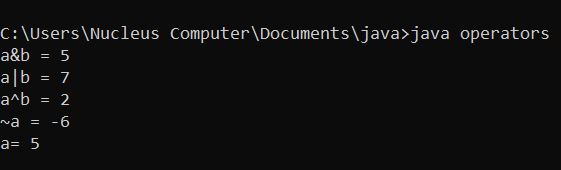
*System.out.println("a^b = " + (a ^ b));*

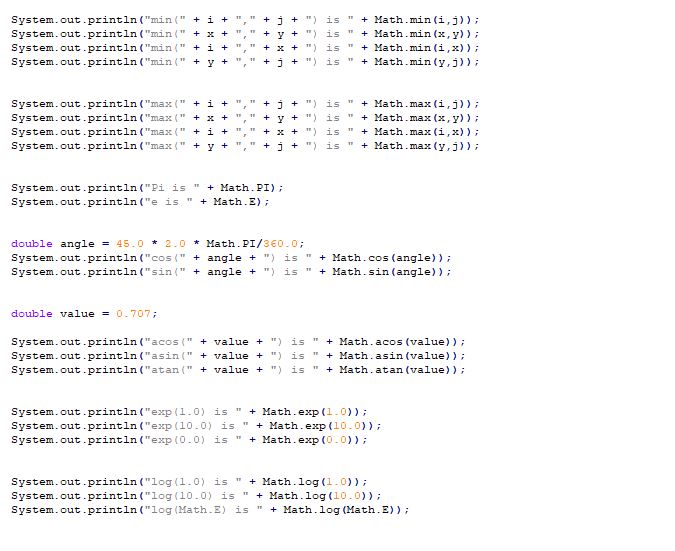
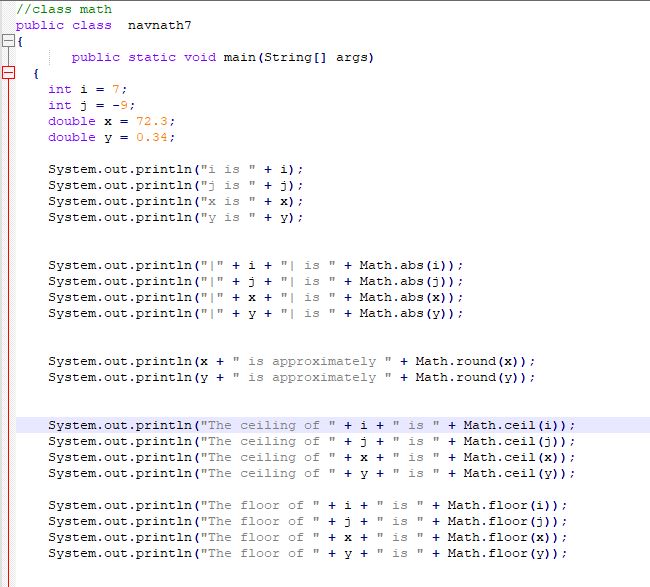
*System.out.println("~a = " + ~a);*

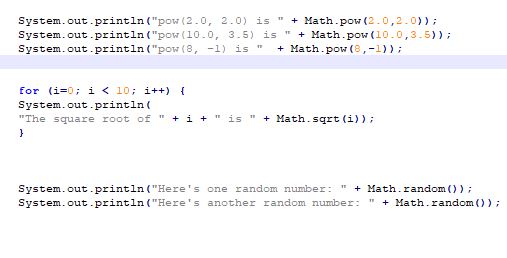
*a &= b;*

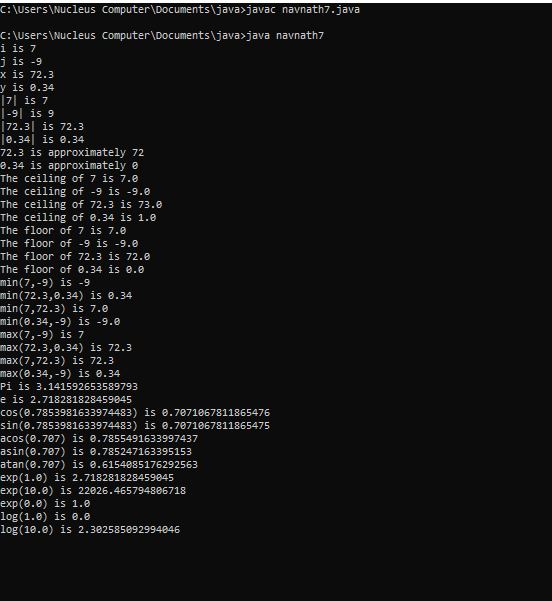
*System.out.println("a= " + a);*

*}*

*}*

**

**

**

*public class navnath8 {*

*public static void main(String args[]) {*

*String s="Sachin";*

*System.out.println(s.toUpperCase());*

*System.out.println(s.toLowerCase());*

*System.out.println(s);*

*System.out.println(s.trim());*

*System.out.println(s.startsWith("Sa"));*

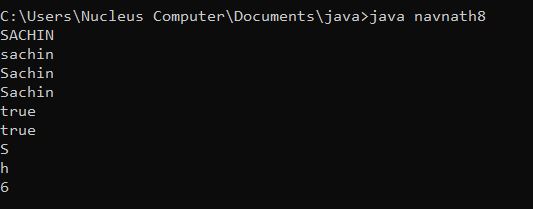
*System.out.println(s.endsWith("n"));*

*System.out.println(s.charAt(0));*

*System.out.println(s.charAt(3));*

*System.out.println(s.length());*

*}*

*}*

*public class Sum {*

*public int sum(int x, int y) { return (x + y); }*

*public int sum(int x, int y, int z)*

*{ return (x + y + z);*

*}*

*public double sum(double x, double y)*

*{ return (x + y); }*

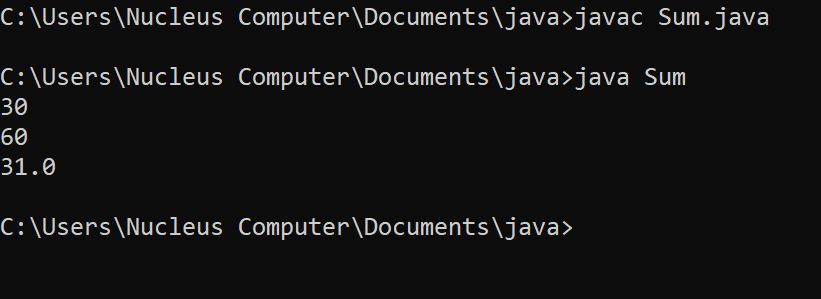
*public static void main(String args[]) {*

*Sum s = new Sum();*

*System.out.println(s.sum(10, 20));*

*System.out.println(s.sum(10, 20, 30));*

*System.out.println(s.sum(10.5, 20.5)); }*

*}*

*class Student6{*

*int id;*

*String name;*

*Student6(int i,String n){*

*id = i;*

*name = n;*

*}*

*Student6(Student6 s){*

*id = s.id;*

*name =s.name;*

*}*

*void display(){System.out.println(id+" "+name);}*

*public static void main(String args[]){*

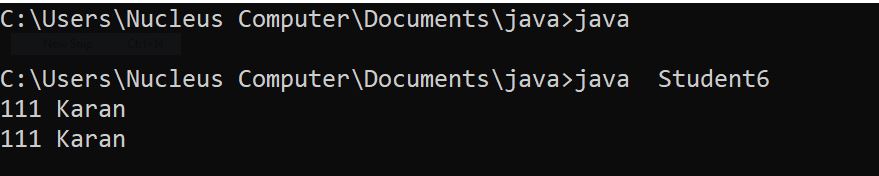
*Student6 s1 = new Student6(111,"Karan");*

*Student6 s2 = new Student6(s1);*

*s1.display();*

*s2.display();*

*}*

*} *

*class Student{*

*int id;*

*String name;*

*Student(int i,String n){*

*id = i;*

*name = n;*

*}*

*void display(){System.out.println(id+" "+name);}*

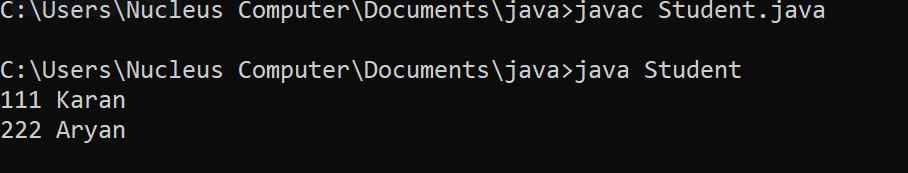
*public static void main(String args[]){*

*Student s1 = new Student(111,"Karan");*

*Student s2 = new Student(222,"Aryan");*

*s1.display();*

*s2.display();*

**

*class array{*

*public static void main(String[] args) {*

*// create a 2d array*

*int[][] a = {*

*{1, 2, 3},*

*{4, 5, 6, 9},*

*{7},*

*};*

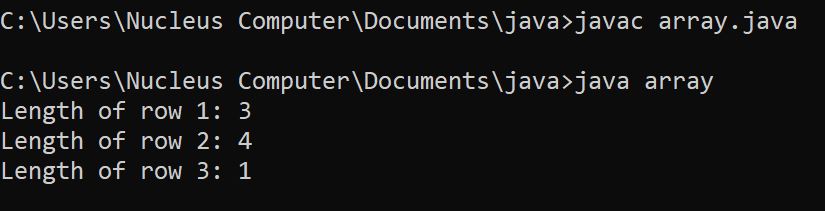
*// calculate the length of each row*

*System.out.println("Length of row 1: " + a[0].length);*

*System.out.println("Length of row 2: " + a[1].length);*

*System.out.println("Length of row 3: " + a[2].length);*

*}*

*}*

*class Employee{*

*float salary=40000;*

*}*

*class Programmer extends Employee{*

*int bonus=10000;*

*public static void main(String args[]){*

*Programmer p=new Programmer();*

*System.out.println("Programmer salary is:"+p.salary);*

*System.out.println("Bonus of Programmer is:"+p.bonus);*

*}*

*}*

**

*class Bike{*

*void run(){System.out.println("running");}*

*}*

*class Splender extends Bike{*

*void run(){System.out.println("running safely with 60km");}*

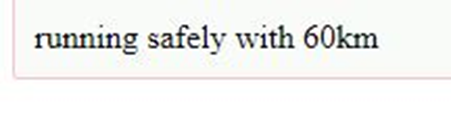
*public static void main(String args[]){*

*Bike b = new Splender();*

*b.run();*

*}*

*}*

**