H. P. D. Jain College, Ara

Principal & Director: (Dr.) Narendra Kumar

Department of B. C. A.

Coordinator: (Dr.) Ashok Prasad

Veer Kunwar Singh University, Ara

Hounorable V. C.: Prof. (Dr.) K. C. Sinha

Subject: Java Programming
Semester: 5th Semester
Paper Name: 5BCA2

(Subject chapter prepared for Online Teaching & Learning Process)

Chapter Name: Introduction to Java

By,

Mr. Ashutosh Pawan

(Faculty)

B.C.A. Department, H. P. D. Jain College, Ara

Some Message to the Students

The content is prepared so that you can understand the programming concepts in java easily hence the language of the content may be different that the books written by respected authors. I have tried my best to cover all the topics related to History and Development of Java, Characteristics of Java Programming, Difference between Java and C and C++ programming language, How to code, compile and run Java programs, Keywords, operators, control statements – selection constructs and loop constructs, nested loops, jump statements – using break and continue as labeled statements, arrays and some other related primary concepts of java.

Some words like class, abstract class or interface or package may have come in this study material that will be discussed and made clear in my next study material.

Teaching and learning process includes preparing fair copy of the subject. Hence my suggestion is that you should also read the reference books appearing in the list of syllabus that you all will have with you since the time of admission in the course. Trying to copy, paste and run the program given in the study material may generate such errors that you cannot understand and debug easily. Hence my suggestion in this respect is to type the program yourself after learning the concept after reading the theory of the topic and then reading and understanding the program. Some example programs have been written in this study material that will certainly generate given type of errors. I have also mentioned how to correct such program.

In your course, you have to prepare assignments, hence I may ask to submit a copy of the topics made available to you online by me and some other questions that I will make available to you online at H. P. D. Jain College, Ara website.

Any query related to my online subject study material can be sent to me directly on the email: diet.bca.hpd@gmail.com. Suggestions and other study material related opinion may be sent directly to me on the email: sug.ashu.bca.hpd@gmail.com.

Depending on the length of the queries I will prepare next online study material in which your problems will be solved. I am trying my best to create online video tutorials also. When it will be ready to upload you will be informed. Visit college & University website regularly and your own whatsapp group to stay updated. Your suggestion will help to improve my material and video tutorial quality or to understand me that I am well prepared before creating online study material or video tutorial.

Be safe, keep hygiene and follow all the guidelines provided by the college, university and the government.

Thank You.

INTRODUCTION

JAVA is an object – Oriented programming language. It is general purpose object- oriented programming language. JAVA was developed in 1991. It is the first technology of fully integrate programming language and services like threads, sockets, GUI components etc. So, that we can built programs that can run on a wide range of different types of hardware architecture and many types of operating system making the language platform independent.

HISTORY OF JAVA

JAVA was developed by team of programming at “SUN MICROSYSTEM” in 1991 by a team of five members : James Gosling, Patrick Norton, Chris Warth, Ed Frank, and Mike Sheridan. JAVA creator James Gosling did not design JAVA or the internet. His objective was to create a common development environment for consumer of electronic device such as Microwave Oven and Remote Control which was easily portable from one device to another. JAVA was initially called “OAK” but was renamed as JAVA in 1995 (according to the myth that the team of members developing the language used to have a coffee named Java, according to another myth it is said that the language was developed at the continent Java hence it is renamed as Java).

IMPORTANT FEATURES OF JAVA

The inventors of JAVA wanted to design a language which could offer solution to some of the language not to be only reliable, portable and distributed but also simple compact and interactive.

Some important features of JAVA programming are----

1. SIMPLE: Java is a simple programming language that helps to do complex program easily. Its simplicity can be understood in the terms that it uses most of the syntaxes of C and C++ except some of the misleading concepts of both the languages. Also Java

considers everything as a part of an object. It uses long descriptive identifier names for classes, methods, literals etc describing the purpose. One can easily identify which identifier is related to which element of Java Program.

A. The names of classes contain first letter of each word capital and remaining letter small.
B. The names of methods contain first letter of each word except first word in the name capital and remaining letter small.
C. The names of literals contain each letter capital.

This makes java simple to learn.

2. PLATFORM INDEPENDENT: Java is a platform independent programming language. Platform independent means program written on one platform can be run easily on another platform. Due to this platform independent feature, we can conclude that a Java program is "write once, run anywhere".

   For Ex. :- If we write a Java program on Windows O.S. it can be easily run on another platform like - Unix, Linux, Mac etc. Java is a platform independent programming language due to bytecode that is generated after successful compilation of java program. [bytecode, an intermediate language, can be executed on any another platform having different set of operating system or different set of hardware architecture which has JVM(Java Virtual Machine).]

3. OBJECT ORIENTED LANGUAGE: Object oriented is an approach to program organization and development that attempts to eliminate some of the drawbacks of conventional programming method by incorporating the basis of structured programming features with several powerful new concepts. The following features of object oriented programming are:

(a) Class: A class is a user – defined data type used to represent a template (a set of properties and methods that operate on them) for defining several similar type of object. Class follows hiding principle (especially for the properties or data that was the main reason for the development of object oriented programming language), so that unauthorized access by the other set of object can be restricted.

(b.) Object: An object is a basic building block of object oriented programming. Object corresponds to real time or run time implement.

An object is a unit of software consist of three characteristics :-

i.) State or Attributes

ii.) Behavior or Function

iii.) Identity

The "State or attributes" (implemented as data in the program) refers to the built-in characteristics of an object. Whereas "Behavior or operations" also called in java as methods, of an object refers to its action.

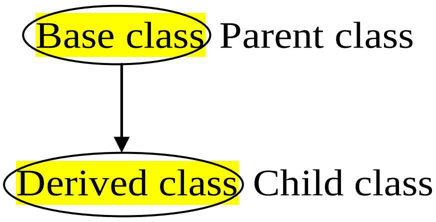
For Example— for nano car:- Brand name, gears, clutch, break, engine capacity, fuel tank capacity, mileage are its state/attribute whether driving is its behavior.

(c.) Data Abstraction: The internal details of the objects are hidden which makes them abstract from external use and this technique of hiding internal details in an object is called data abstraction. The keywords private, protected and public plays an important role in this context. On the other hand Java has another visibility level known as default.

(d.) Encapsulation: Grouping of data and function together into a single unit through specific mechanism refers to encapsulation. We can use the encapsulated object using the same unit in which they are encapsulated. class provide the mechanism of encapsulation of data together with the methods (member functions) into a single unit.

Encapsulation can hide data and functions from classes to other classes.

(e.) Inheritance: If a method of deriving new class from existing class/ classes. Inheritance actually provides the concept of reusability of existing code without repeating them in the program. The existing classes or the class defined earlier in the program or in a package is known as parent class or base class or super class and the class being defined currently to extend the existing features is known as the child class or derived class or sub class.



(f.) Polymorphism: The term polymorphism is made up of two terms "Poly" means *many* and " Morphism" means *forms*. It provides a common interface to carry out similar tasks. In simple, we can say that one name, many duties. Polymorphism is a useful concept in OOPs language.

JVM(Java virtual machine) is an environment that makes polymorphism possible during runtime of the program. It is a software which can be used to execute a bytecode.

(g.) Robust: Java is a Robust(strong) language due to object oriented features. We cannot implement or execute any program code in java outside the class except package and import statements which can be written at the very top of the java program if necessary.

(h.) Secure: Java is a secured object oriented programming language because each and every code should be written within a class.

SOURCE Program →Java compiler →Byte code →
Executor code ^ Machine code/ JVM

Java Program Demonstration

Simple Java Program

```
class  Test1
{
public static void main(String args[])
      {
           System.out.print(“H. P. D. Jain College Online
           Tutorial”);
       }
 }
```

Before briefing about the above example I want to suggest everyone that tradition make easy to work and learn. Hence as explained about the simplicity of java you should take a look of the tradition of java. In above example

class: class is a keyword used every time to define a class, every java executable program must contain at least one class specification.

Test1: It is an identifier name that refers the name of the class being defined in Java program. The name starts with capital letter and contains rest letters in small and a numeric digit 1 in the end. (Identifiers may be opted independently following the rules of Identifier naming convention of Java).

public: public is access specifier or visibility level indicates that the method (i.e. main method) associated with class Test1 is a public member that is accessible from anywhere.

static: It is a keyword indicates that this method (i.e. main method) associated with class Test1 and only one existence of the member of that class is created.

void: void is a keyword used to specifies the return type of method(member function/ function). If we specify the void keyword, it indicates that this method doesn’t return anything to that part from where the method has been called.

main(): During the runtime of Java program, first of all an object corresponding to the class that contains main() method, is loaded by the Java interpreter with the help of JVM in computer's memory and thereafter the execution of Java program codes starts from the main() of the class. Java main() can receive command line arguments, if sent at the runtime to the program, that are stored in the String array parameter, here args.

System.out.print: To produce output or massage from a java program at runtime, java program code contains it in a statement. Here *System* is a predefined class in java, *out* is a predefined object (objects are always created for concrete classes in java, *out* is an object created for PrintStream class defined in *java.io* package), *print* is a method defined in PrintStream class that accepts a String object and writes the string on Java program execution screen or at command prompt.

Difference between C++ and Java :--

(a.) Java doesn't support operator overloading.

(b.) Java doesn't have a template classes, friend classes and functions as in C++.

(c.) Java doesn't support the concept of multiple inheritance this can be done using a new feature called interface.

Interface is like a class that contains one or more declared method.

(d.) Java doesn't support global variables every variables and method is declared within the class.

(e.) Java doesn't use pointer. To achieve runtime polymorphism where derived class defines or overrides the same signature of method as defined in parent class, reference variable is used in place.

(f.) Java has replace the destructor function with the help of finalize(). The programmer need not define finalize() method in every Java program as Java garbage collector perform required destruction of unused objects just before quitting from the execution of Java program.

(g.) There is no header file in Java. In place Java uses *package* that contains predefined classes and interface that further contains literal and methods. Concept of preprocessor directives and preprocessor operators has been left in java.

(h.) Java has not included struct and union in its keyword list. goto and const are reserved words of Java and programmer cannot make use of these two words.

JAVA CLASS LIBRARY:-

Java provides a rich set of class libraries that provides a large number of classes and interfaces by defining them in relevant packages that any java program can use it. These libraries are used for Input/Output, mathematical, networking, event handling and many other capabilities. These class library summarize the primary function. A package is a collection of classes and interfaces. The various java class libraries/packages are given below:-

(1.) java.io: This package supports all types of input/output operations.

(2.) java.applet: This package support to built applet program that executes with the help of appletviewer or in web browser as an HTML code.

(3.) java.net: This package supports to implement networking through java program.

(4.) java.util: This package supports the java utilities.

(5.) java.lang: This package supports java functionalities. All the classes and interfaces defined in this package are imported by default. *System* is one such class defined in java.lang package.

(6.) java.beans: This package supports java software components.

(7.) java.awt: This supports to design abstract windows toolkit (awt) to construct GUI environment.

A. java.awt.event :- This package supports to handle events for awt component.

B. java.awt.image :- This package supports to perform image processing.

STRUCTURE OF JAVA PROGRAM:-

Primary Java program starts with the construction of classes and class contains data members and methods that operates on the data combined together in that class. However, the following structure shows a java program structure performing any designated task:

1. Documentation section
2. package section
3. import section
4. interface section
5. class definition section
6. class containing main method

```
{
  main method definition
  {
      -----------------
      -----------------
   }
}
```

e.g. :- / ** A program to display a message */

```
import java.lang.*;
 class test
  {
      public static void main(String args[])
      {
         System.out.print(“ first java program”);
```

```
        }
    }
```

(1.) Documentation section :- This section usually contains the comment lines used to specify the name of the program and other details. This section is optional.

a. /*-------------------*/(Multiline comment)

b. //---------------------- (Single line comment)

c. /**------------------------ */

It is used for document file, which store statements of this line in another file. Document file is useful for user and is not compiled or interpreted at run time. But when the applet programs are run using appletviewer jdk tool giving the name of the java file an html file is automatically created.

(2.) Package section: This section is used to declare the package name that is used to inform the computer that classes or interfaces defined in the program belongs to the specified package. This section is optional. But remember the package statement should be the first executable statement of java program file.

(3.) Import Section: This section is used to import the one or more classes or interfaces of specified package. The import statement may be understood as including a header file in 'C' or 'C++' program. This section is optional. Every java program imports the classes and interfaces defined in java.lang package by default.

(4.) Interface Section: This section is used to define any interface be declaring one or more abstract methods. Actually every declared method in interface body are abstract by default. Literals as final variables can also be declared inside interface body. This section is optional. Interface is just like fully abstract class. But are useful to achieve multiple inheritance.

(5.) Class Section: The class is the primary and essential element of a java program and the java program file may contain multiple class definition. These classes are used to the objects. If the java program is created to run at least one class in the java program file must contain main() defined as in above example.

(6.) Main method Definition class: The main method specifies the entering point of a java program. This is essential part of a java program. This is used to create objects of required classes and also used to establish communication between them.

CODING JAVA PROGRAM:

Open a text editor (notepad or msdos editor) to create a file that contains the program named as First1.java. The file name should match the class that we define. Java is case sensitive so be careful while entering both the name of file and the name of class defining main() method.

COMPILING AND RUNNING A JAVA PROGRAM

Compilation: Before executing a Java program, The program must be compiled using the javac jdk tool as shown below:

Step 1: Save the file in specific directory

e.g. – D:\java program\Test1.java

Step 2: Then open the DOS command prompt.

Step 3: Then go to my computer → C drive → program files → JDK(java development kit) → Bin → Then select the path and copy the path.

Step 4: In DOS command prompt type or paste the copied path as the statement given below----

```
C:\HPDJain\>path=”C:\program files\Java\JDK\Bin”  (press enter)
```

Step 5.- Then press enter key. The above work enable javac compiler at command prompt.

Step 6: Now change the directory prompt by using msdos ‘cd’ command to the directory in which java program has been saved.

```
C:\HPDJain\> cd d:\java program (press enter)
```

Step 7: Then compile the program by using javac compiler.

```
D:\java program\> javac Test1.java (press enter)
```

Step 8: It will create the bytecode for the compiled java program which is platform independent.

Step 9: Run the program with the help of java interpreter (jdk tool) by providing name of the class in which main() method has been defined.

```
D:\java program\> java Test1 (press enter)
```

Some note should be kept in mind while using above steps to run other programs of java.

Once you have followed the steps 1 through 6 and save every java program in specified directory and msdos prompt is still opened before compiling and running other java program, you need not to follow all the steps through step 6.

JDK tool comes inside Java Development Kit after successful installation of java in user’s computer memory.

Java Compiler $\xrightarrow{\text{invoked on}}$ Java Program $\xrightarrow{\text{generates}}$ bytecode

Java Interpreter $\xrightarrow{\text{invoked on}}$ bytecode $\xrightarrow{\text{transferred to}}$ JVM for execution

JVM is an environment (application software) which convert bytecode into executable code.

VARIABLES:

The variable is the basic unit for accessing computer's temporary storage through a java program. We all know anything can be stored in computer's memory. The value of a variable can vary during the period of execution of java program. A variable is defined by the combination of an identifier and an optional initializer. Also one thing should be remembered in java that java is strictly typed language, hence any statement of Java program should not use any variable that has not been initialized or any value has not been stored in declared variable.

Declaring a variable:

Variables must be declared before use in java program. The syntax of variable declaration is:

Syntax: Datatype var1, var2, ---- varn;

Where, Datatype is the type of data. var1,var2, ,var n are the list of variables.

e.g. String stdname;
int mark1, mark2 ,
mark3; float avg;

Rules for naming a variable/Identifier:

1. It must begin with alphabet, capital or small.
2. It cannot contain special symbols and blanks spaces.
3. A keyword cannot be not used as a variable.
4. Long identifier names should be avoided.

Data type:-

1. Predefined
   a. Numeric
   b. Non numeric
2. User defined
   a. Class
   b. Array
   c. Interface

Numeric

1. Integer

   |

   a. byte - 1 byte
   b. short - 2 bytes
   c. long - 8 bytes
   d. int – 4 bytes

2. Float

   |

   a. float – 4 bytes
   b. double – 8 bytes

Non- Numeric

1. char – 2 bytes
2. boolean – 1 bit

KEYWORDS USED IN JAVA:--

abstract

throw

const(reserved)

boolean

finally

continue

int

native

if

else

volatile

private

final

while

float

public

interface

return

long

byte

double

static

new

implement

char

synchronized

instance of

default

catch

switch

import

class

this

throws

short

super

package

extends

for

do

void

until

break	try	protected
tangent	goto (reserved)	

Some Important Points :

- JVM (Java Virtual Machine) is an environment (software) which converts bytecode into executable code.
- Escape Sequence literals starting with '\' character are used for formatting output statement. It is the combination of two character e.x.: \n,\t,\a and so on but give single character.
- Dot operator (.) is known as membership operator and is used to access members of class or objects.
- Instance is created for a class.
- instanceof () is the special type operator which returns the information either an object belongs to the given class or not. It returns either *true* or *false*.

```
e.g. – Class   Test
                 {
                    --------
                    ---------
                  }
                    Test t1,t2,t3; //reference variable declaration in Java
             boolean b = t1.instanceof (Test);
```

- finalize() is used at the place of destructor. Programmer need not define this method in every java program.
- If we want to use another language program code to our java program this process is done with the help of native keyword.
- import is used in the program to use the classes or interfaces defined in package.
- volatile is a keyword used to create a variable which stores data temporarily.
- super keyword is used to call constructor of base class or to use overridden methods.

- static has different use for:
  - ➢ declaring a variable that remain sharable among objects and is created in a single copy in memory.
  - ➢ defining methods that can be accessed by the class name.
- implements keyword is used with interface to achieve multiple inheritance in java program.
- extends keyword is used to inherit any one class in new class being defined.
- If any method is not defined but declared in a class, the method and the class should precede abstract keyword.
- boolean is used to hold true or false.

❖ LITERALS: LITERALS are ‘C’ or ‘C++’ language like constants and is a fixed value which doesn’t vary during the program execution. It can be divided into two types------

(i) Numeric constant

(ii) Non - numeric constant

❖ OPERATOR: An operator is a symbol or group of characters that tells the computer to perform certain mathematical and logical manipulation and is used to result some value. It is used in program to manipulate data and variables.

In java different types of operator is used which are

(i) Arithmetic operator

(ii) Logical operator

(iii) Relational operator

(iv) Bitwise operator

1) Arithmetic operator: Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. The basic arithmetic operators supported by java are:

+ (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus), ++ (Increment), -- (Decrement) and = (Assignment).

2) Logical operator: && (and), || (or), and ! (Not) are logical operators used in java. Relational expressions are the operands of logical operators.

3) Relational operator: The relational operator determines the relationship between two operands or data items. The outcome of these operations is a boolean value. = = (exactly equal to), != (not equal to), > (greater than), < (less than), >= (greater than equal to), <= (less than equal to) are relational operators.

4) Bitwise operator: The bitwise operator is used for the bitwise logical decision making. Bitwise operator can be applied to the integer types, long, int, short, char and byte. The following bitwise operators used in java are:

(a.)	<<	(Shift left)
(b.)	>>	(Shift right)
(c.)	>>>	(Shift right zero fill)
(d.)	<<<	(Shift left zero fill)
(e.)	~	(Unary complement)
(f.)	|	(Bitwise inclusive OR)
(g.)	^	(Bitwise exclusive OR)

Except above operators java facilitates Comma operator, Ternary operator (also called Conditional operator) and instanceof (type of comparison operator)

❖ Conditional Statement :

Control flow statements or Conditional statement control the flow of execution to advance and branch based on changes to the state of program. Java language uses following Conditional statement:

(i) Selection statement
(ii) Iteration statement
(iii) Jump statement

Selection statement include Simple if, if-else, Nested if-else, switch statement etc. They are also known as Selection construct. Iteration statement is also called Loop constructs which include while, do - while, for loop etc. break statement is used to terminate the execution of switch or loop statement. A continue statement causes control to be transferred directly to the conditional expression that controls the loop. In java break and continue can be used as Labeled statements to transfer the execution to a certain part of the program, if necessary.

Q: W.A.P. to print name and address.

Sol^{n}.

```
class    Test3
 {
    public static void main (String args[])
     {
         System.out.println(“H. P. D. Jain College”);
          System.out.prinln(“Station Road, Ara”);
     }

 }
```

Q: W.A.P. to add two numbers.

Sol^{n}:

```
class        Add
{
  public static void main(String args[])
  {
        int a, b, sum;
        a = 5;b = 8; sum = a+b;
         System.out.println(“ sum of two number=” +sum);
   }

}
```

If input is required from keyboard, one such java program looks like below:

```
 import java.util.Scanner;
 class   TestInput
 {
    public static void main(String args[])
        {
        Scanner sc=new Scanner
             (System.in) ; int a,b, sum;
          System.out.print(“ Enter first no.=”);
                a = sc.nextInt();
           System.out.print(“ Enter second no.=”);
                 b = sc.nextInt(); sum=a+b;
            System.out.println(“Sum of two number=”+sum);

            }

     }
```

Java program can take command line input i.e. when the interpreter is invoked to run java program, the user can send inputs to the program. Also a program is developed to process any command line input then it is the responsibility of the user to send the same number of inputs separated by blank spaces to the program otherwise the program abnormally terminates or exceptionally terminates. In such a situation Java Runtime Environment should know about the exceptional termination of the program so that it can handle the exception occurred. To illustrate this the above problem is solved below that processes command line argument:

```
class  CmdLineInput         {
            public static void main(String  args[]) throws Exception
            {
                    int  num1,num2,sum;
                    num1= Integer.parseInt(args[0]);
                    num2= Integer.parseInt(args[1]);
                    sum = num1 + num2;
                    System.out.println(num1 + “+” + num2 + “=” + sum);
            }
          }
```

Integer and Exception are Java Classes. parseInt() is a static method defined in Integer class.

Q: W. A. P. to check which number is greater between two numbers.

Sol^n:

```
import  java.util.Scanner;
class   FindGreater
 {
        public static void main(String  args[])
        {
                int  num1, num2;
                Scanner  sc= new Scanner(System.in);
                System.out.print("Enter first number:\t");
                num1 = sc.nextInt();
                num2 = sc.nextInt();
                if (num1 >= num2)
                System.out.print("First number is greater");
                else
                System.out.print("Second number is greater");
        }
}
```

Q: W. A. P. to display the day name for a day number.

Sol^n:

```
class  MonthName
{
        public static void main(String args[])
        {
         int   dn=1; /*change every time the number, save the
         program, compile and then run to get the relevant day
         name*/
        if(dn==1)
        System.out.print("Sunday");
        else if(dn==2)
        System.out.print("Monday");
        else if(dn==3)
```

```
            System.out.print(“Tuesday”);
            else if(dn==4)
            System.out.print(“Wednesday”);
            else if(dn==5)
            System.out.print(“Thursday”);
            else if(dn==6)
            System.out.print(“Friday”);
            else if(dn==7)
            System.out.print(“Saturday”);
            else
            System.out.print(“No such day exist”);
        }
        }
```

Above program using switch – case selection construct:

Sol^{n}:

```
class  MonthName              {
 public static void main(String args[])    {
  int   dn=1; /*change every time the number, save the
        program, compile and then run to get the relevant day
        name*/
switch (dn) {
 case 1:      System.out.print(“Sunday”);        break;
 case 2:      System.out.print(“Monday”);        break;
 case 3:      System.out.print(“Tuesday”);       break;
 case 4:      System.out.print(“Wednesday”);  break;
 case 5:      System.out.print(“Thursday”);      break;
 case 6:      System.out.print(“Friday”);        break;
 case 7:      System.out.print(“Saturday”);      break;
 default:     System.out.print(“No such day exist”); break;
 }
}
}
```

Q: W.A.P. to add digits of a number.

Sol^n:

```
import  java.util.Scanner;
 class     Digit

 {
      public static void main (String args[])

      {
              Scanner sc=new Scanner(System.in);
              int  a, sum;
              sum=0;
              System.out.print(“enter a no.=”);
              a = sc.nextInt();
               while (a!=0)
               {

                       sum= sum + a % 10;
                       a = a / 10;
               }

               System.out.println (“\n Sum of digits=”+sum);
      }

}
```

Q: W.A.P. to return reverse of a number and display whether the reverse and the number are same or not.

Sol^n:

```
import  java.util.Scanner;
 class       CheckReverse

{
     public static void main (String args[])
     {
             Scanner sc=new scanner (System.in);
             int a, num, sum, rev;
                   sum = 0, rev = 0;
               System.out.println(“Enter a number=”);
                    a=sc.nextInt();
                    num = a;
                    while(a!=0)
                    {
```

```
                                sum = a % 10; a = a / 10;
                                 rev = rev * 10 + sum;
                                 }
  System.out.println("\ Reverse of a number="+rev);
  if(num = = rev)
  System.out.println("Number and Reverse are same");
  else
  System.out.println("Number and Reverse aren't same");
}
}
```

Q: W. A. P. to print the numbers from 1 to 80. The loop condition will terminate the program at 100.

Sol^{n}:

```
class  UseBreak
{
        public static void main(String args[])
        {
        int num=1;
        do
                {
                if(num>80)
                break;
                System.out.print(num++ + "\t");
                }while(num<=100);
        }
}
```

Q: W. A. P. to print only even numbers between 1 to 100 using continue statement and incrementing the variable by 1.

Sol^{n}:

```
class UseContinue
{
        public static void main(String args[])
        {
                for (int  i=1; i<=100 ; i++)
                {
```

```
                if (i %2 !=0)
                        {
                                i++;
                                continue;
                        }
                System.out.print(i+ “\t”);
        }
}
```

Q: W. A. P. to print all prime numbers between 1 to 100.

Sol^{n}:

```
class  AllPrime100
{
        public static void main(String args[])
        {
                int  num,i;
                boolean  prime;
                for(num=1;num<=100;num++)
                {
                        prime=true;
                        for(i=2; i< (num/2); i++)
                        {
                        if(num%i==0)
                        prime=false;
                        }
                        if(prime==true)
                        System.out.println(“Prime number\t”+num);
                }
        }
}
```

Q: Write a program to illustrate labeled statements in java.

Sol^{n}:

```
class  AllPrimeTo100
{
        public static void main(String args[])
        {
                int  num,i;
                boolean  prime;
        outer:      for(num=1; ;num++)
                    {
                        prime=true;
                    inner:      for(i=2; i< (num/2); i++)
                                {
                                if(num%i==0)
                                { prime=false;
                                  continue  outer;}
                                if(num>100)
                                  break  outer;
                                }
                    if(prime==true)
                    System.out.println("Prime number\t"+num);
                }
        }
    }
```

Arrays in Java:

Array is a group of related data items stored in contiguous memory location associated with each other having only one name.

Declaration of Array :--

Before using an array variable in any java executable statements, it should be declared using following syntax:

Syntax :

```
        Datatype  variable_name [ ];

                Or,

         Datatype  [ ]
   variable_name ; Eg :-    int a [ ];
```

In the above declaration the array don’t mention the size & memory space is created during runtime of the program. Actually in Java everything is an object, array is also an object hence it should be instantiated first.

Creating memory space :--

After declaring the array new keyword is used to allocate memory space for that array.

Syntax :-

```
    Variable_name = new   datatype [size];
```

```
    Eg :- a= new int [3];
```

The above example can be combined by the following statement. We can say that

Syntax :-

```
      Datatype Variable_name [ ] = new datatype [size];

      Datatype [ ] Variable_name = new datatype [size];
```

Initialization of Array :-

After declaring array and creating memory space required for array, it can be used to store values for what it has been created. One such process is known as initialization. (Array variables can also take inputs from the users.) Every time a value is accessed from the array, subscript is associated with it as:

```
             Array_name [ ] = value;
```

```
Eg :-        num [0] = 35;

             num [1] = 40;

             ---------------------
```

```
-------------------
------------------
num [n] = 19;
```

Note : Every array element is associated with an index starting with 0 and ending at size - 1.

Unlike C, Java protects array from overflow and underflow. Trying to access an array bound its boundaries will generate an error message.

We can also initialize array automatically in the same way as the ordinary variables when they are declared.

Datatype array_name [ ] = { list of values };

The array initializer is a list of values separated by commas and surrounded by curly braces.

Eg :- int num [ ] = {35,40,20,57,19};

ARRAY LENGTH

As stated earlier array is an object hence it has a property named length that provide the size of the array. We can obtain the length of array using array_name.length.

The array_name.length returns or gives the information about how much element can be stored in the array. Eg :- int a size = a.length

Q: Write a program in Java to store 10 numbers and display their sum.

Sol^{n}:

```
class TestArray
{
public static void main(String args[])
{
int num[] = new int[10], sum;
```

```
        int i;
        for(i=1;i<=9;i++)
        {
        num[i]=num[i]+1;
        }
        for(i=0;i<10;i++)
        {
        sum=sum+num[i];
        }
        System.out.print(“Numbers are\n:”);
        for(i=0;i<10;i++)
        System.out.print(num[i]+ “\t”);
        System.out.print(“\nSum=” +sum);
        }
}
```