Java[J2SE] Docs 5/12/2018

N Hemanth

CONTENTS :

1. About JAVA
2. Features of JAVA
3. Drawbacks of C and C++ that are drawn to java major advantage
4. JVM Architecture
5. Description for JDk/bin folder files

Java is simple, secure , object-oriented ,class-based general purpose programming language.

**Java Applications:**

Java run under over 3 billion devices world wide.it have many applications.

* Desktop applications
* Web applications
* Enterprise applications
* Mobile
* Embedded System etc

**Types of Java Applications** :

1. Standalone applications : it is used to develop Desktop applications mostly run traditional operating systems ex: music player
2. Web applications : it is used to develop server and run dynamic web pages it uses servlets,JSP,springs,hibernate etc
3. Enterprise application : it is used to develop enterprise or industrial applications for this it will use EJB[Enterprise Java Bean]
4. Mobile application : it used to develop small mobile applications.

**Java Platforms/Editions :**

1. J2SE[Java Standard Edition]
2. J2EE[Java Enterprise Edition]
3. J2ME[Java Micro Edition}
4. JavaFX[for Internet applications]

**Features of Java :**

1. Simple
2. Object-oriented
3. Secured
4. Portable
5. Robust
6. Platform independent
7. Interpreted
8. Architectural neutral
9. High performance
10. Multi-threaded
11. Distributed
12. Dynamic
13. **Simple** :

→ java syntax based on c++.

→ java removed complicated features like explicit pointers ,operator overloading.

→ java has automatic garbage collector which removes automatically unreferenced

Objects.

**2. Object-oriented:**

→ everything in java is object , which is like a blueprint for class.

**3. Security : java is known for its security**

→ no explicit pointers

→ java runs inside a virtual machine

→ class loader : it provides security by separated package from classes.

→ Bytecode verifier : java checks weather bytecode is clean or not

→ java provides security.

**4.Platform independent:**

→ java is portable and platform independent write once and run anywhere.

→ for c and c++ it produce machine based .exe files if this files run under another system

It wont work

→ for java where it carries bytecode to any level and can make it work

**5. Robust :**

→ it had strong memory management

→ there is automatic garbage collector

→ type checking and exception handling.

**6.Portable:**

→ java will allow you to carry bytecode to any level and execute on all machines which

Have JVM

**7. High-Performance :**

→ java is faster than traditional interpreted programming languages but it is still slower

Than c and c++.

**8.Distributed :**

→ java is distributed languages which can obtain buy Rmi and ejb allows to transfer files

Destination

**9. Multi-Threaded :**  java supports thread concept which can all web and server side a

Applications

**10.Architectural neutral :**

→ if it is 32 bit or 64 bit pc java has a sample data typing..

**11. Dynamic :**

→ java supports dynamic class loading it also supports native class loading.

-------------------------------------------------------------------------------------------------------------------------

**Drawbacks of c and c++:**

* C and C++ are platform dependent.while java is platform dependent.
* They are used mainly for system programming but java supports wide range of applications including internet.
* C and C++ has no bound checking
* Security issues were high while compared to java
* Complexity of code increases while writing large software.
* C is not a object oriented and c++ supports but its not true
* Both uses pointers which leads to security fixes

--------------------------------------------------------------------------------------------------------------------------

**JVM Architecture :**

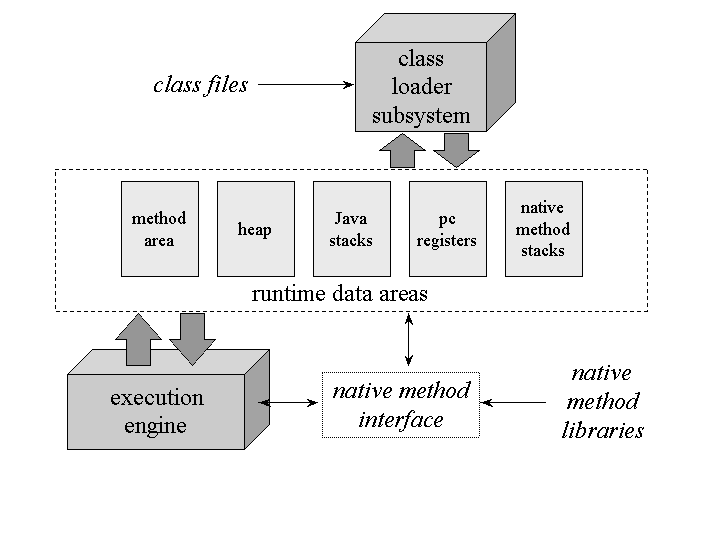
Jvm is a abstract machine where bytecode is executed. Jvm is available on various hardware and software platforms

**Implementation :**

jvm implements is known as JRE[java runtime environment].

**RuntimeInstance :**

to run the java class every time a jvm instance created.



**1.Class loader subsystem:**

class loader subsystem under jvm loads .class file into it it had mainly three types.

**Bootstrapclassloader :**

bootstrapclassloader is super class of extension class loader . it loads rt.jar files which contain all apis of j2se like java.lang,java.util,java.math etc.

**Extension class loader :**

extension class loader is child of bootstrap class and parent for system/application class loader which loads jar files from $java\_Home/jre/lib/ext directory.

**System/Application class loader :**

it is a child class of extension class loader it loads all class files under class path. When every programs executed it point to present working directory.

**2.class or method area :**

Class or method area stores class structures such as runtime constant pool method data, fields and code fro methods.

**3.Heap:**

It is runtime area where objects were allocated.

**4.stack :**

Stack store frames , it holds local variables and partial result , it is also play key role in method invocation and return.each thread has a private jvm stack created at the same of creation of thread.a new frame is created when every time a method invoke and destroy after its completion .

**5.PC Registers :**

Pc registers holds addressed of currently executing on jvm .

**6. Native stack methods :**

It all contains all native methods used in application .

**7.Execution engine :**

execution engine where programs executes it contain mainly three parts

1. **Virtual processor**
2. **Interpretation :**

where it reads bytecode streams and executes it

1. **JIT-Compiler:**

it transforms instruction of jvm into cpu instruction, also it reduce the code which have similar functionality

**8.Java native Interfaces :**

Java native interface is a framework where it helps to communicate with other programming languages , it uses OS libraries from console input and Output.

-------------------------------------------------------------------------------------------------------------

**JAVA/BIN FILES:**

**Basic Tools:**

AppletViewer:

Applet viewer is used to use applets outside of browser

Extcheck:

Extcheck is used for solve the jar files version conflicts.

Jar:

Create and manage java and archive files[jar]files.

Apt:

Apt is for annotation processing tool

Java:

Launcher for java application , a single launcher for both development and deployment is used for an old .jre is no longer exist

Javac :

Compiler for java programming language

Javadoc:

API documentation generator

Javah:

C header and stub generator , used to write native methods.

Javap:

Class file disassembler

Jdb:

The java debugger.

**Security tools:**

These security tools help you set security policies on your system and create applications that can work within the scope of security policies set at remote sites.

Keytool:

Manages keystores and certificates.

Jarsigner:

Generates and verify JAA signatures.

Policytool:

GUI tool for managing policy files.

These security tools help you obtain, list, and manage Kerberos tickets

Kinit:

Tool for obtaining Kerberos v5 tickets .

Klist:

Command line tool to list entries in credential cache and key tab

Ktab:

Command line tool to help user manage entries in key user table.

**Internalization Tools:**

This tool helps to create localizable applications.

Native2ascii:

Convert text to unicode latin-1

**Remote Method Invocation [RMI] Tools :**

These tools help to create applications that interact over the Web or other network**.**

Rmic :

Generate stubs and skeletons for remote objects.

Rmiregistry:

Remote object registry service

Rmid:

RMI activation system daemon

Serialver:

Return class serialVersionUID.

JAVA IDL and RMI-IIOP Tools :

These tools are used when creating applications that use OMG-standard IDL and CORBA/IIOP.

Idlj:

Idlj generates java bindings from IDL files.IDL[interface definition language] is a generic term that lets a program or object to communicate with another program written in another unknown language. To use CORBA functionality

Orbd:

Provides support for clients to transparently locate and invoke persistent objects on servers in the CORBA environment. ORBD is used instead of the Transient Naming Service, tnameserv. ORBD includes both a Transient Naming Service and a Persistent Naming Service. The orbd tool incorporates the functionality of a Server Manager, an Interoperable Naming Service, and a Bootstrap Name Server. When used in conjunction with the servertool, the Server Manager locates, registers, and activates a server when a client wants to access the server.

Servertool:

Provides ease-of-use interface for the application programmers to register, unregister, startup, and shutdown a server.

Java Deployment Tools:

Utilities for use in conjunction with deployment of java applications and applets on the web.

Javafxpackager :

Packages javafx applications for deployment.  
Pack200:

Transforms jar file into compressed pack200 using the java gzip compresser , the compressed pack files are compressed JARs which can be dirctly deployed saving bandwidth and reducing download time.

Unpack200:

Transforms a packed file of pack200 into jar file.

Java Web Start Tools:

It used for conjunction with java web start

Javaws :

Command line tool for java web start and setting various options

Java Troubleshooting,Profiling,Monitoring and Management Tools:

Jcmd :

JVM diagnostic commands tool - sends diagnostic command request to run a java virtual machine.

Jconsole:

A JMX -compliant graphical tool for monitoring java virtua, machine.it can monitor both local and remote jvms it cal also monitor and manage an application .

Jmc:

Java mission control client includes tools to monitor and manage java application without introducing performance overhead normally associated with these tools.

Jvisualvm :

A graphical tool that provides detailed information about the Java technology-based applications (Java applications) while they are running in a Java Virtual Machine. Java VisualVM provides memory and CPU profiling, heap dump analysis, memory leak detection, access to MBeans, and garbage collection

Java Web Services Tool :

Schemagen :

Schema generator for java architecture for xml binding.

Wsgen:

Tool for generate JAX-WS portable artifacts.

Wsimport:

Tool for generate JAX-WS portable artifacts

Xjc :

Binding compiler for java architecture for xml binding.

Day-2

DataTypes and Operators

DATA TYPES:

In java they are mainly two types of data types.  
1.Primitive Data Types

2.non-primitive data types

1.primitive data types:

1. Numeric : byte,short,int,long
2. Floating-point : float and double
3. Character : char
4. Boolean - boolean

.

2.non-primitive data types:

String and array :

-----------------------------------------------------------------------------------------------------------

Byte : it's a 8-bit two's bit 2’s complement integer, it's value range between -128 to 127

Its default value is 0

Programs : Test byte range value by increasing its range and see output if it shows error write the error.

public void bytechecking() {

byte a = 128;

System.out.println(a);

}

Output : Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Type mismatch: cannot convert from int to byte

at java\_prac.DatatypesandOperators.bytechecking(DatatypesandOperators.java:17)

at java\_prac.DatatypesandOperators.main(DatatypesandOperators.java:24)

Short: it’s a 16-bit two’s complement integer, its value range between -32,768 to 32,767, its default value is zero.s

Programs : Test short range value by increasing its range and see output if it shows error write the error.

public void shortrange() {

// TODO Auto-generated method stub

short b = 32768;

System.out.println((int)b);

}

Output : Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Type mismatch: cannot convert from int to short

at java\_prac.DatatypesandOperators.shortrange(DatatypesandOperators.java:23)

at java\_prac.DatatypesandOperators.main(DatatypesandOperators.java:31)

Int :

Int is a primitive data type its a 32 bit two’s complement number its value ranges from

- 2,147,483,648 to 2,147,483,647

Programs : Test int range value by increasing its range and see output if it shows error write the error.

public void intrange() {

int c = 2147483648;

System.out.println(c);

}

Output :

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The literal 2147483648 of type int is out of range

at java\_prac.DatatypesandOperators.intrange(DatatypesandOperators.java:28)

at java\_prac.DatatypesandOperators.main(DatatypesandOperators.java:36)

Long :

Long is a 32 bit 2’s complement it value range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 .

Program : Test program without oyt writing l in in input

public void longrange() {

long l = -9223372036854775808; // with out ‘l’

System.out.println(l);

}

Output : Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The literal 9223372036854775808 of type int is out of range

at java\_prac.DatatypesandOperators.longrange(DatatypesandOperators.java:32)

at java\_prac.DatatypesandOperators.main(DatatypesandOperators.java:41)

Program : test the long range value by increasing its value .

public void longrange() {

long l = -922337203685477581000l;

System.out.println(l);

}

Output :

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The literal 922337203685477581000l of type long is out of range

at java\_prac.DatatypesandOperators.longrange(DatatypesandOperators.java:32)

at java\_prac.DatatypesandOperators.main(DatatypesandOperators.java:41)

Program : type cast long range maximum value to int

Output : for positive it will give -1 and negative its -1

-----------------------------------------------------------------------------------------------

**EXERCISE TASKS :**

**Exercise-1:**

int x=10;

x++;

sopln(x);

Program:

public void exe\_1() {

//POST-INCREMENT OPERATION PROGRAM

/\* x value will be incremented with one value, its a post increment operation it initialize

\* variable first and increment with one\*/

//Declared a variable x as 10 for post increment operation

int x = 10;

x++;

System.out.println(x); //output : 11

//output of this program is 11

}

--------------------------------------------------------------------------------------------------------------

**Exercise-2:**

int a=10,b=20,c=30;

b=a;c=b;

sopln(c);

int a=b=c=10;

sopln(c);

int a,b,c;

a=b=c=10;

sopln(c);

Program :

public void exe\_2() {

/\*equal assignment operation first initialize

\* three Integer variables with value later

\* assign variable with another check the operation\*/

//Declared and initialize variables with values

int a = 10,b=20,c=30;

b=a;

c=b;

System.out.println(c);

int a=b=c=10;//duplicate local variable ‘a’

System.out.println(c);

int a,b,c;// all are duplicate local variable.

a=b=c=10;

System.out.println(c);

/\* java will throw java.lang.Error duplicate value a two times, duplicate value b one time and c two time

\*/

}

Programm - 3:

private void exe\_3() {

// TODO Auto-generated method stub

//Initializing a character a

char ch = 'a';

//incrementing a character

ch++;

System.out.println(ch); // output = b

/\* output of this program will be = b

\* because its use character alphabet values its

\* will print next alphabet automatically.

\* its increment of character will be next alphabet

\*/

}

Program - 4 :

public void exe\_4() {

//initializing a value double of 10.5

double d = 10.5;

d++;// incrementing a double with one its add 1 to 10.5

System.out.println(d);// output = 11.5

/\*double value will be incremented with one

\* it will become 11.5

\*/

}

Program - 5:

public void exe\_5() {

//declaring a boolean flag with true

boolean b = true;

b++;//trying to increment a boolean

/\*it will throw java.lang.Error :unresolved compilation

\* error. type mismatch : cannot convert from boolean

\* int.because boolean is data type with 1 bit it will

\* have only true either false we cannot perform any arithmetic operation.

\* it will used for true or false condition.

\*/

}

program -6:

public void exe\_6() {

//initializing a byte with value 20

byte b = 20;

/\*duplicate a local variable b and type mismatch

\* cannot convert from int to b

\*/

byte b = b+1;

/\*duplicate local variable b

\* but mismatch error will be solved

\*/

byte b = (byte) (b+1);

System.out.println(b);

/\*two types of compilation errors will be seen here

\* first one is duplicate local variable there must

\* be no declaring variable that already declared.

\* second one is type mismatch from into to byte because

\* initially in integer data types its initially all declared as int

\* if u perform any increment arithmetic operations it will show error

\* we have to explicit cast of byte it will tell to java to ignore

\* bits that won't fit, they will be cut even if it changes the value of a number.

\*/

}

Program -7 :

public void exe\_7() {

//initializing two byte variables

byte a = 10;

byte b = 20;

/\*adding two byte variables to byte c

\* it will show typemismatch cast error tells

\* us to explicit cast to byte

\*/

byte c = a+b;

/\*adding two byte variables with explicit cast ,

\* but compiler will show duplicate local c variable.

\*/

byte c = (byte)(a+b);

System.out.println(c);

/\*duplicate local variable c and cast error of type conversion

\* we have to explicitly cast the byte to resolve the issue.

\*/

}

Program - 8 :

public void exe\_8() {

//dividing a value with zero and printing the output

System.out.println(10/0);

System.out.println(10/0.0);

/\*java compiler throw java.lang.Arithmeticexception

\* value cannot be divide with zero in both cases.

\*/

}

Program - 9 :

public void exe\_9() {

System.out.println('a'+'b'); // output = 195

System.out.println('a'+1); // output = 98

System.out.println('a'+1.2); //output = 98.2

/\* IN first println statement we are adding two character a and b

\* it will add its ascii values, ascii value of a = 097 and b = 098

\* ...in second println statement it will print ascii value of = 97 +1

\* its gives output of 98

\* in third println statement it will add ascii value of a with 1.2

\* gives output 98.2

\*/

}

Program - 10

public void exe\_10() {

//initializing string ashok

String a = "ashok";

//initializing three int variable with values.

int b=10,c=20,d=30;

/\*adding all three int values and trying to assign in string variable

\*/

a = b+c+d;

//adding all int values and assigning in a int variable

a = a+b+c;

/\*adding a string variable with int values. it show cast error

\* cannot convert from string to int

\*/

b = a+c+d;

/\*in first compilation error we can't store it in int beacuse string is immuatble it cannothanhge

\* as well as we assign int to string instead on concatenating

\* in second compilation error the string will concatenation with int values but

\* it can't store in a a int its cast error.

\*/

}

Program - 11:

public void exe\_11() {

System.out.println(10<10.5);//output = true

System.out.println('a'>100.5);//output = false

System.out.println('b'>'a');//output = true

System.out.println(true>false);//compilation error

/\* in first three print statements it operation works successfully output

\* will five true or false for condition case

\* but in third scenario > operator cannot done on true or false

\* it can't determine the value and throw and compilation error.

\*/

}

Program - 12:

public void exe\_12() {

//all four print statements are condition based.

System.out.println(10 == 20);//output = false

System.out.println('a' == 'b');//output = false

System.out.println('a' == 97.0);//output = true

System.out.println(false==false);//output true

/\* In first print statement condition failed because 10

\* is not equal to 20.

\* In second print condition failed because a is not equal to b

\* in third print statement the condition will be true because the ascii

\* value of a is 97 97==97.0 the condition is true.

\* in final fourth print condition statement we campring

\* false with false it true both values are same.

\*/

}

Program - 13:

public void exe\_13() {

//instantiated thread object t1

Thread t1 = new Thread();

//instantiated another thread object t2

Thread t2 = new Thread();

Thread t3 = t1; // assigning t1 object to newly created thread variable t3

System.out.println(t3); // output = Thread[Thread-0,5,main]

System.out.println(t1);// output = Thread[Thread-0,5,main]

System.out.println(t2);// output = Thread[Thread-1,5,main]

System.out.println(t1==t2);//comparing t1 object with t2 gives false

System.out.println(t1==t3);//comparing the t1 with t3 gives true

/\* we created two separate thread objects t1 and t2 both will have separate thread values

\* if we compare it will give false,but in third we just instancited thread we assign the t1

\* value to the t3 variable if we compare both have same value condition satisfied and it will

\* print true.

\*/

}

Program - 14 :

public void exe\_14() {

System.out.println(true&false); // output = false

System.out.println(true|false); // output = true

System.out.println(true^false); // output = true

/\*in first we performed a bitwise and on and gate table 1 & 0 = false

\* in or gate table 1 | 0 or true | false = true

\* in third bitwise xor or in xor gate 1 ^ 0 = true.

\*/

}

Program -15 :

public void exe\_15() {

System.out.println(4&5); // output = 4

System.out.println(4|5); // output = 5

System.out.println(4^5); // output = 1

/\* binary value of 4 = 0100

\* binary value of 5 = 0101

\* bitwise and (&)=

\* ------------------------------

\* 0100 = 4

\* -------------------------------

\* binary value of 4 = 0100

\* binary value of 5 = 0101

\* bitwise or (|)=

\* ------------------------------

\* 0101 = 5

\* -------------------------------

\* binary value of 4 = 0100

\* binary value of 5 = 0101

\* bitwise xor (&)=

\* ------------------------------

\* 0001 = 1

\* -------------------------------

\*/

}

Program - 16 :

public void exe\_16() {

System.out.println(~true); // compilation error

System.out.println(~4); // output = -5

/\*it will given java.lang.Error negation operators

\* is undefined for boolean types

\* for second print statement it will give -5 negation will applied.

\*/

}

Program - 17:

public void exe\_17() {

System.out.println(!false); // output = true

System.out.println(!4); //compilation error

/\*in first statement not operator will successfully apply on boolean case

\* where as it can't apply to integer , it shows ! operator undefined for type

\* int

\*/

}

Program - 18 :

public void exe\_18() {

int x = 10, y = 15;

if (++x < 10 || ++y > 15) {

x++;

} else {

y++;

}

System.out.println(x+"----"+y); // output (12,16)

int x = 'a';//compilation error

System.out.println(x); // output = 12

/\* In if condition we have do a logical or operation in this case

\* either one of the statements must be true.

\* in if +++x = 11 < 10 condition fails another side ++y = 16 > 15

\* condition true, it will enter into if condition perform

\* increment operation on x giving final value 12

\* and y = 16

\* compilation errors is creating a local duplicate variable.

\*/

}

public void exe\_18B() {

int x = 10, y = 15;

if (++x < 10 & ++y > 15) {

x++;

} else {

System.out.println("ok");

y++;

}

System.out.println(x+"----"+y); // output (11,17)

// int x = 'a';//compilation error

System.out.println(x); // output = 11

/\* Bitwise and in if condition it will take the

\* boolean values in if condition and operates

\* bitwise and operation in above it has true and false

\* it give false so else state executed.

\*/

}

public void exe\_18C() {

int x = 10, y = 15;

if (++x < 10 | ++y > 15) {

x++;

} else {

System.out.println("ok");

y++;

}

System.out.println(x+"----"+y); // output (12,16)

// int x = 'a';//compilation error

System.out.println(x); // output = 11

/\*bitwise or operator in above if condition

\* it will take resultant boolean values and on them

\* bitwise or operation swill perform true and false in or

\*/

}

Program - 19 :

public void exe\_19() {

int x = 130;//initializing a integer variable

byte b = (byte) x; //explicit type casting int to byte

System.out.println(b); //output = -126

/\* byte memory has range o f -128 to 127 but int value 130 when we typecaste

\* it it will cut of the memory and print the bits.

\*/

}

Program - 20 :

public void exe\_20() {

int x = 150;

short s = (short)x;

byte b = (byte) x;

System.out.println(s); // output = 150

System.out.println(b); // output = -106

/\*short ranges for -32,768 to 32,767, the int x value is 150

\* is within range of shorthand by explicit casting short error and with

\* in range it will print as same

\* but coming to byte casting it int value is exceeds the byte range

\* so by explicitly type casting it will reduce the memory seize and

\* value inside will change by bit manipulation.

\*/

}

Program - 21

public void exe\_21() {

double d = 130.456;

int x = (int) d;

System.out.println(x);//output = 130

byte b = (byte) d;

System.out.println(b); //output = -126

/\*when we type cast a double to int it will remove decimal values

\* and it will print.

\* in byte scenario it same but exceeding memory range it will give a another value.

\*/

}

Program -22 :

public void exe\_22() {

int x = (10>20) ? 30:((40>50)?60:70);

System.out.println(x); // output = 70

/\* in above statement we done multiple ternary operations

\* we checked with 10 > 20 if condition success it store 30 else

\* it will lead to another statement in that we compared 40 > 50 condition

\* failed it store the values of 70.

\*/

}

Program - 23 :

public static int m1(int i) {

/\* we created static method so there is no need to instance the class

\* we declared method m1 with return type int

\* so the method must return a integer value to the calling

\* method with parameter int i

\*/

System.out.println(i);

return i;

public static void main(String[] args){

System.out.println(m1(1)+m1(2)\*m1(3)/m1(4)\*m1(5)+m1(6));

}

}

Output:

1

2

3

4

5

6

12

DAY - 5

1.Write a java program to print the following series using while loop and for loop

1,2,4,7,11,16,22…. For a given number n.

Program :

public void number\_series() {

/\*1.Write a java program to print the following

series using while loop and for loop

1,2,4,7,11,16,22…. For a given number n.\*/

//Using for loop

int n = 25;

int i;

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

int num = (i\*(i-1)/2+1);

System.out.print(num+",");

}

System.out.println(" ");

int j=1;

//using while loop for sequence

do {

System.out.print((int)(j\*(j-1)/2+1)+",");

j++;

} while (j<=n);

/\*output of program = 1,2,4,11,16,22,......\*/

}

Output : 1,2,4,11,16,22,......

2.Write a java program to print all the alphabets using while loop and for loop.

Program :

public void alphabet\_series() {

/\*Write a java program to print all the

\* alphabets using while loop and for loop.

\*/

//using for loop.

char ch = 'a';

System.out.println("Using For loop:");

for (int i = 0; i < 26; i++) {

System.out.print(ch++);

}

//using while loop

System.out.println("using while loop:");

char alph = 'a';

do {

System.out.print(alph);

alph++;

} while (alph<='z');

//output : a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z.

}

Output : a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z

3.Write a java program to print the following pattern

Input n = 3

\*\*\*\*\*

\*

\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program :

public void start\_sequence() {

/\*Write a java program to print the following pattern

\*Input n = 3

\*/

// \*\*\*\*\*

// \*

// \*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*

// \*\*

// \*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int n = 3;

//Using for loop

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

int ul = i\*5;

for(int j = 0;j<ul;j++) {

System.out.print("\*");

}

System.out.println(" ");

for(int k=0;k<i;k++) {

System.out.print("\*");

}

System.out.println(" ");

for(int l =1;l<=ul;l++) {

System.out.print("\*");

}

System.out.println(" ");

}

//Using while loop

//output :

// \*\*\*\*\*

// \*

// \*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*

// \*\*

// \*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}