Question Answers:-

Q1) what is meant by precedence of operators?

Ans) The order in which operators of an expression are going to be evaluated and resolved in a predetermined order is called operator precedence.

Q2) What is a literal?

A literal is a sequence of characters (digits, letters and other characters) used in a program to represent a constant value. (to be stored in variables)

**Not to be written if the question carries a small mark:** java language specifies five major types of literals. They are:

* Integer literals
* Floating\_point literals
* Character literals
* String literals
* Boolean literals

Each of them has a type associated with it. The type describes how the values behave and how they are stored.

Q3) State the java concept that is implemented through:

1. a superclass and a subclass
2. the act of representing essential features without including background details.

Ans) i) Inheritance (here, subclass inherits the superclass)

ii) Abstraction (here, we only represent the essential features of an object and don’t consider the irrelevant details of an object)

Q4) State the differences between a constructor and a method.

Constructor and normal methods both are actually methods of an object (runtime instance of a class. And if we consider the definitions of them both definitions are part of a class). But there are some differences between them.

**First difference:** constructor of a class essentially has the same name as of the class. (i.e. if a class is named as **Student**, then the name of the constructor is also **Student**) but a function (normal method) cannot have the same name as of the class.

**Second difference:** Constructors necessarily do not have return types. Whereas a function must have a return type.

**Third difference: Constructors** are invoked automatically at the time of object creation.

Consider the example: Suppose, we have written a class named “Student” and then we are creating an object of the class “Student” named std1.

Then the object std1 must be created like this:

Student std1=new Student();

New Student() invokes the default constructor of the Student class.

Whereas, methods or functions need to be called.

Q5) Write statements to show finding the length of a character array char[] differs from finding the length of a string object str.

Ans) For array , length is a variable so we use Char.length and for String object, (String is a class within java.lang package) length() is a method so we use str.length().

(Note: for String class, a method length() exists of return type int (integer) which returns the length of the String)

Q6) Give one example each of a primitive datatype and a composite data type.

Ans) int, short, char, double, float : those are primitive datatype. (i.e. in-build datatype)

Classes and arrays are the examples of composite datatype.

Q7) give one point of difference between unary and binary operators.

Ans) Operators that act on a single operand are called unary operator and operators that act on two operands are binary operators.

**Example**: Unary operator: - unary minus

int x=25;

y=-x;//this minus is unary minus, negating the value of x, then assigns the //value to y (negated value of x)

Binary operator: - binary minus

int x,y,s;

x=25;

y=20;

z=x-y;//this minus operates on two operands and returns the result of the //subtraction

Q8) what is ASCII value? (It is not a question from text paper. But somehow it is a good question related to the syllabus)

Ans) The **American Standard Code for Information Interchange** a character encoding scheme originally based on the English alphabet that encodes 128 specified characters - the numbers 0-9, the letters a-z and A-Z, some basic punctuation symbols, some control codes that originated with Teletype machines and a blank space - into the 7-bit binary integers.

**Important ASCII code:**

**ASCII value Characters**

48-57 0-9

65-90 A-Z

97-122 a-z

Q9) what is operator associativity? (Another question which is not from the question papers but, a good question for exam)

Ans) Associativity is the rule of evaluating an operator first among more than one operators of same precedence (or same priority level).

Associativity defines in the case that there are several operators of the same priority or precedence level, which one must be evaluated before the rightmost one or the leftmost one.

Q9) what is a token in java?

Ans) The smallest unit of a java program is known as token.

(In simple terms, a java program is a collection of tokens, comments and white spaces.)

Java language includes five types of tokens:-

* Reserved keyword (like: this, abstract, const, final etc.)
* Identifiers
* Literals
* Operators
* Separators

Q10) what is an identifier?

Ans) Identifier is programmer-designed token. Those are used for naming classes, methods, variables, objects, labels, packages and interfaces in a program. An identifier can be thought of symbolic name given by the programmer to classes, methods, variables etc.

(Not to be written if the question comes for a small marks:

Java identifiers follow the following rules:

1. They can have alphabets, digits and underscore and dollar sign characters (**Can’t have white space**)
2. They must not begin with a digit
3. Uppercase and lowercase letters are distinct
4. They can be of any length.

)

Q11) State the difference between token and identifier

Ans) The smallest unit of a java program is known as a token

Whereas, symbolic names given by programmer (i.e. programmer-designed tokens) to variables, objects, classes, interfaces, labels, packages etc. are known as identifiers.

Q12) what is an operator in java?

Ans) An operator is a symbol (a type of java token) that takes one or more arguments and operates on them to produce a result.

**Example:** binary operator “+”, new, & etc.

Q13) what is a separator in java**?**

Ans) Separators are symbols (another type of java token) used to indicate where groups of codes are divided are arranged. They basically define the shape and function of our code.

**Example:** parentheses (“()”), braces (“{}”), semicolon, comma etc.

Q14) what is a constant in java?

Ans) Constants in java refer to fixed values that do not change during the execution of a program.

Java supports several types of constants:

1. Numeric constants
2. Integer constants
3. Real constants
4. Character constants
5. Character constants
6. String constants

Q15) what is a variable in java?

A variable is an identifier (programmer designed token that is used for naming purpose) which denotes a storage location used to store a data value. Unlike constants that remain unchanged during the execution of the program, a variable may take different value at different times during the execution of a program.

Q16) Explain instance variable. Give an example.

Ans) The attributes (data members basically) of an object are represented by variables are known as instance variables.

Instance variables are the data members of a class which are needed to maintain differently for different instances (runtime instances or objects) of a class.

E.g. consider an object of a class Bankaccount, the instance variables for the object can be:

* Balance
* AccountType ( i.e. is it saving bank account, or recurring bank account or fixed deposit or some other options)
* ClientName
* Clients\_id (i.e. bank account number)

Etc.

Q17) explain class variable. Give an example.

Ans) The data members of a class or variables of a class, which are needed to be maintained entirely for the class (i.e. needed to be maintained entirely for all instances of the class) , are called class variables.

E.g. :- Consider the above example of a class Bankaccount. Now, suppose, the no. of accounts which is to be created from a particular branch of a bank is restricted, then we need to maintain a variable entirely for all instances of Bankaccount created from that branch of the bank. Then we are in need of a class variable. It could be accountcount of integer type variable.

Q18) Explain local variables.

Ans) Local variables are variables which are declared and used within methods of a class and which are local to that method (in terms of scope of those variables).

Local variables can also be declared inside program blocks that are defined between an opening brace and a closing brace.

Q19) We often define symbolic constants for a class or for a method of a class which appear repeatedly in a number of places in that method with the help of “final” keyword. What are the facilities of defining such symbolic constants (suppose, no Pi method exists in the java.lang.Math class. Then we need to define a symbolic constant named Pi for using it frequently in some method)

Ans) facilities of defining such symbolic constants:-

* **Modifiability:-** Suppose, we do not define pi as a symbolic constant. We just use the value of pi whenever we need it to calculate something in a method (Since, symbolic constants are defined for a method) . Now, if we may like to change the value of pi from 3.142 to 3.14159 to improve its accuracy. Then we need to change the value all those places of a program where we use it. But, if we define a symbolic constant named pi, then we need to change the value only once (where, we assign the value to symbolic constant pi) . Now, you might be thinking why do we need to use the final keyword to define symbolic constant . What will happen if we just define a variable named pi. The answer is we are using the final keyword for protecting pi from accidental modification because, pi’s value meant to be constant for the whole method where it is used frequently.
* **Understandability:-** When a numeric value appears in a program, its use is not always clear, especially when the same value means different things in different places. For example, the number 50 may mean the number of students at one place and the ‘pass marks’ in some other place of the same program of the same method. We may forget what a certain number meant, when we read the program some days later. But, if we associate the value with some identifier and define it as a symbolic constant then the chance of forgetting the purpose of using a numeric value in a program is eliminated.

Q20) what are the types of casting shown by the following examples?

1. double x=15.2;

int y=(int)x;

1. int x=12;

long y=x;

Ans) First one is explicit type casting. And the second one is implicit type casting.

Q21) Name a java keyword that:

1. indicates that a method has no return type
2. Stores the address of the currently-calling object.

Ans) First one is **void** and second one is **this**. (When a function has return type as void, that means the function is not returning something and **this** indicates the current instance of the class i.e. indicates the currently-calling object)

Q22) What is an exception?

Ans) This is the anomalous (unexpected) situation which occurs during the execution of the program (i.e. at runtime)

Q23) what does a class encapsulate?

Ans) A class encapsulates the data members and the member functions which perform some operations on it.

The data members (attributes) of a class represents the state of an object and the member functions associated with the data members of the object represents the behaviour of an object.

Q24) Differentiate between call by value and call by reference or pass by reference.

Ans) in the case of call by value, the actual parameters are copied to the formal parameter. (i.e. when we call the function, using some arguments the arguments’ values are copied as the formal parameters) where as in the case of call by reference, the actual parameters are not copied to the formal parameters. They are just referred to by another name or same name even (in case of same name, i.e. if the data members of the class and the parameters to a method of that class have same name, then they could be differentiated using **this** keyword). In java, there is no concept of passing by value. Everything is passed as references. But in languages like c, c++ call by value concept is supported. (in c: only call by value and in c++ both call by value and call by reference)

**Another difference:**

In case of call by value, memory is allocated separately for actual and formal parameter.

But in case of call by reference, the memory is not allocated separately for actual and formal parameters.

Q25) write a java expression for √(2as+u2)

Ans) Math.sqrt(2\*a\*s+Math.pow(u,2))

Q26) Name the type of error (syntax, runtime or logical error) in each case given below:

1. Division by a variable that contains a value of zero.
2. Multiplication operator used when the operation should be division
3. Missing semicolon

Ans) now, division by a variable that contains a value of zero, does not cause any syntax or compilation error (syntax error and compilation time error both are same) but it will cause runtime error.

Multiplication operator used when the operation should be division: - this does not cause any compilation error as well as runtime error (program will compile as well as run perfectly) the purpose for writing the program will not solved. So, it is a logical error in the program.

Missing semicolon is a syntax error (will be generated at the compilation time)

Q27) Create a class with one integer instance variable. Initialize the variables using:-

1. Default constructor
2. Paramterized constructor

Ans) suppose, the class’s name is Example and it has only one instance variable named x of integer datatype. Then the default constructor will look like:

public Example()

{

x=0;

//you can initialize it to other value

}

And the parameterized constructor will look like:

public Example(int x1)

{

x=x1;

}

Or you could write the parameterized constructor as:

public Example(int x)

{

this.x=x;

}

Where, this keyword differentiates between the data member which is to be initialized) and the parameter send to the constructor.

Q28) complete the code below to create an object of Scanner class

Scanner sc=\_\_\_\_\_\_\_\_\_ Scanner (\_\_\_\_\_\_\_\_\_\_\_\_);

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

Q29) What is an array? Write a statement to declare an integer array of 10 elements.

Ans) Array is a group of related data items of same type that are referred by a common name.

Or array is a collection of contiguous memory locations to store homogeneous elements (elements that belong to same datatype) and is referred by a common name.

int IntArr[]; //declaring an array of integer datatype which will only store integers

IntArr=new int[10];//creates the slot such that IntArr can hold 10 integers

Or, you could do;

int[] IntArr; //declaring an array of integer datatype which will only store integers

IntArr=new int[10];//creates the slot such that IntArr can hold 10 integers

This int[] IntArr is also a valid syntax for declaring an array of integer type. Or you could do;

int[] IntArr =new int[10];

Or,

int IntArr[]= new int[10];

All these four syntaxes are valid syntaxes.

Q30) Name of the search or sort algorithm that:

1. Makes several passes through the array, selecting the next smallest item in the array each time and placing it where it belongs to the array or not.
2. At each stage, compares the sought key with the key value of the middle element of the array.

Ans) Ans of a) selection sort

Ans of b) Binary search

Q31) differentiate between public and private modifiers for members of a class.

Ans) Private members of a class are accessible only within the class. It can only be accessed via the member functions of the same class within the class definition (i.e. within class). But it cannot be accessed by other classes or within other classes via the objects of the class (which contains those data members)

Whereas public members of a class are available to all the classes of all the packages. Other classes can access public members of a class via the objects of that class.

Q32) what are the values of x and y when the following statements are executed?

int a=63, b=36;

boolean x=(a>b)?true:false;

int y=(a<b)?a:b;

Ans)

**Background knowledge for solving the program:** The ?: operator is called ternary operator or conditional operator. The general form of conditional operator is as follows:-

Conditional expression? expression1:expression2

The conditional statement is evaluated first. If the result is true, expression1 is evaluated and returned as the value of the conditional expression. Otherwise expression2 is evaluated and its value is returned.

For example, the segment

if(x<0)

{

flag=0;

}

else

{

flag=1;

}

Can be written as

flag=(x<0)?0:1;

Now solve the given questions.

Ans of first one) true will be assigned to boolean variable x

Ans of second one) 36 will be assigned to integer variable y.

Q33) State the values of n and ch.

char c=’A’;

int n=c+1;

char ch=(char)n;

Ans) int n=c+1;

Now here c will implicitly converted to integer (it has ASCII value as 65. So, it will be converted to 65) . now n is clearly 66.

char ch=(char)n;

Now, the value of n is explicitly type casted to character. (it cannot be done automatically by the compiler. So, we need to typecast it explicitly) so, ch will be a character whose ASCII value will be 66. So, ch will be ‘B’.

Q34) what will be the result stored in x after evaluating the following expression?

int x=4; x+=(x++)+(++x)+x;

Ans) the value of the expression is solely depended on the associavity of the binary plus operator (“+”)

Now, it has the associavity from left to right.

Now, x+=(x++)+(++x)+x means

x=x+(++x)+(x++)+x

first x’s value is 4. Then (++X) is first increased then we get its value

so, x=4+5+5+6

x=20 (Note that x++ i.e. first its value is evaluated then it is increased.

Q35) Give the output of the following program segment:

double x=2.9,y=2.5;

System.out.println(Math.min(Math.floor(x),y));

System.out.println(Math.max(Math.ceil(x),y));

Now,

static double floor(double a)

This method returns **the largest** (closest to positive infinity) **double value that is less than or equal to**

**the argument and is equal to a mathematical integer.**

i.e. Math.floor(x) will return 2.0

and

static double ceil(double a)

This method returns **the smallest** (closest to negative infinity) **double** value **that is greater than or**

**equal to the argument and is equal to a mathematical integer.**

i.e. Math.ceil (x) will return 3.0

Now,

static double min(double a, double b)

This method returns the smaller of two double values.

i.e. Math.min(Math.floor(x),y)=>Math.min(2.0,2.5) =>2.0  
and

static double max(double a, double b)

This method returns the greater of two double values.

Math.max(Math.ceil(x),y) =>Math.max(3.0,2.5)=>3.0

Q36) State the output of the following program segment

String s=”Examination”;

int n=s.length();

System.out.println(s.startsWith(s.substring(5,n)));

System.out.println(s.charAt(2)==charAt(6));

Ans) **the background knowledge:**

**String** is a sequence of characters. in java, internally Strings are maintained as character arrays.length() is a method of String class.

int length()

This method returns the length of this string .

So, s.length() will return 11 and this value will be assigned to n.

String substring (int beginIndex, int endIndex)

This method returns a new string that is a substring of this string .

substring(5,n) here beginning index is sent as 5 and endIndex is sent as 11. So, the string which will

be returned from the **substring method** is nation (remember, in String, the index is counted from 0. i.e. the starting character of the index has an index as 0. So, the character which has index is ‘n’. In languages like c , after every string, a null character ‘\0’ is included automatically. This denotes the termination of the string. unlike c, in java, no ‘\0’ is added to the string, because of the better index out of bound checking in java.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| character | n | a | t | i | o | n |
| index | 5 | 6 | 7 | 8 | 9 | 10 |

Now,

boolean startsWith(String prefix)

This method tests if this string starts with the specified prefix string.

s.startsWith(s.substring(5,n)) it checks if the string s (which is Examination) is started with the string **“nation”** (which is returned from the substring method) or not. Now, according to the argument, the result will be returned as false. So, false will be printed. (remember, false will be printed, not False)

char charAt(int index)

This method returns the char value at the specified index.

Now, in string, characters are started from the starting index =0;

So, s.charAt(2) will return a and s.charAt(6) will also return a. Now, s.charAt(2)==s.charAt(6) is a conditional expression. This can either be evaluated to true or to false. Here, it will be evaluated to true. So, true will be printed (remember true will be printed, not True).

Q37) State the method that:

1. Converts a string to a primitive float data type.
2. Determines if the specified character is an uppercase character.

Ans) Ans of i) Float.parseFloat() i.e. parseFloat() method of Float class.

Ans of ii) Character.isUppercase() i.e. isUppercase() method of Character class.

Q38) State the data type and values of a and b after the following segment is executed:

String s1=”Computer”,s2=”Applications”;

a=(s1.compareTo(s2));

b=(s1.equals(s2));

Ans) Now, two methods are used of String class.

First one,

boolean equals(Object anObject)

This method compares this string to the specified object.

Second one,

int compareTo(String anotherString )

This method compares two string s lexicographically.

**Background knowledge of lexicographical ordering:** The words in a dictionary are listed in alphabetic, or lexicographic, order, which is based on the ordering of the letters in the alphabets.

The **lexicographic ordering**  on A1 × A2 is defined by specifying that one pair is less than a second pair if the first entry of the first pair is less than (in A1) the first entry of the second pair, or if the first entries are equal, but the second entry of this pair is less than (in A2) the second entry of the second pair. In other words, (a1, a2) is less than (b1, b2), that is (a1, a2) ≺ (b1, b2),

either if a1 ≺1 b1 or if both a1 = b1 and a2 ≺2 b2.

**Now, back into our problem: -**

**a=** (s1.compareTo(s2));

Here, we will compare string s1 with string s2 lexicographically.

First index (i.e. 0th position) of both strings:-

S1.charAt(0) will be compared with S2.charAt(0)

Now, ‘A’ is compared to ‘C’ here.

Now, here, since S1.charAt(0) is unequal with S2.charAt(0) a=s2.charAt(0)-s1.charAt(0)

i.e. a= ‘C’-‘A’;

a=2; (because ‘C’ has the ASCII value 67 and ‘A’ has the ASCII value 65)

and clearly a will be of int type (since the compareTo function of the string class has the return type as int)

Note that, if s1=”Applications” and s2=”Computer”

a=(s1.compareTo(s2));

then a will be -2;

The rule of s1.compareTo(s2) is if s2.charAt(0) ≠s1.charAt(0) then it will return s2.charAt(0)-s1.charAt(0)

And if s2.charAt(0)==s1.charAt(0), then s2.charAt(1) and s1.charAt(1) will be compared. If s2.charAt(1)≠s1.charAt(1) then it will return s2.charAt(1)-s1.charAt(1). Otherwise if they are equal s2.charAt(2) and s1.charAt(2) will compared and so on.

**Note**

And the equals method (of the String class) returns boolean value (either true or false). So, b is of boolean type. Now, it will return false. Since, s1 is not equal to s2.

Q38) What will be the following code output?

String s=”Malayalam”;

System.out.println(s.indexOf(‘m’));

System.out.println(s.lastIndexOf(‘m’));

Ans) Now, the background knowledge required for the problem:

The details of the method indexOf of String class which is used here, is given below:

int indexOf(int ch)

This method returns the index within this string of the first occurrence of the specified character.

Note that, String class has more than one indexOf method for different purposes. But we are only concerned with the method which is used in the given code.

It returns the index (in string, index starts from 0, i.e. the first character of the String has an index 0) of the first occurrence of the specified character. Otherwise (if the character is not present in the string) it returns 0. Now, note that, the method has argument type as int, so the character will be implicitly converted to the corresponding ASCII value.

And the detail of the lastIndexOf method which is used in the given code is given below:

int lastIndexOf(int ch)

This method returns the index within this string of the last occurrence of the specified character.

So, the output of the given code will be:

0

8

Q39) In the program given below, state the name and the value of the

1. Method argument or argument variable
2. Class variable
3. Local variable
4. Instance variable

class myClass

{

static int x=7;

int y=2;

public static void main(String args[])

{

myClass obj=new myClass();

System.out.println(x);

obj.sampleMethod(5);

int a=6;

System.out.println(a);

}

void sampleMethod(int n)

{

System.out.println(n);

System.out.println(y);

}

}

Now, static members of the class are always maintained entirely for the class (i.e. they are not maintained differently for different instances of the class. They are maintained singly for all instances of the class). So, static members are class variables. So, in the given code, static int x: this x is a class variable.

y is the instance variable (maintained differently for different instances of the class) .

a is the local variable. (because it is local to a method definition. Outside the method definition, there is no existence of the variable a)

and n is the argument variable. Because, it is sent as argument to a method.

Q39) State the difference between a class and an object.

Ans) class: static time definition of new type as a collection of data members and associated functions (that perform some kind of operations from which runtime instances called objects can be created. So, we can imagine class as a set of objects that share (i.e. those objects share) common characteristics and behaviour. It is also known as object factory or producer of objects.

Whereas, object is the runtime state of a conceptual framework (that conceptual framework is class) which encapsulates typed data and typed operations that correspond to a real world entity or thing for the purpose of computational modeling.

Q40) what does the token **“keyword”** refer to, in the context of java?

Ans) Keywords are reserve words which convey special meanings to the java compiler (javac).

Example of keywords: void, if , else, break, continue etc.

Q41) State the difference between entry controlled loop and exit controlled loop?

Ans) The loop in which condition is checked first (i.e. the validity condition for the loop to be executed) then the loop body gets executed is called entry controlled loop. They are also called pre tested loop.

e.g.: -while loop, for loop

And the loop in which the loop body gets executed first then the condition is checked is called exit controlled loop. They are also called post-tested loop.

e.g.: -do-while loop

Q42) what are the two ways of invoking functions?

Call by parameters

And call by reference.

(though this is the answer mentioned in the testpaper, I personally doubt this answer. These are the ways of sending parameters)

Q43) what is the difference between / and the % operator.

Ans) / operator operates on two operands and results as the quotient of two operands. i.e. a/b returns the quotient part of the division of a by b and % operator also operates on two operands and results the remainder. i.e. a%b returns the remainder of the division process of a by b.

Q43) State the total size in bytes. Of arrays a[4] of char data type and p[4] of float datatype.

Ans) Array is a collection of contiguous memory location for storing/holding homogeneous elements (of same datatype).

a[4] of char datatype holds four elements of char datatype. Now, in java a char datatype’s variable takes 2 byte. So, total size of a[4] is 8 byte.

And the size of p[4] of float datatype is 16 byte. (a variable of float datatype takes 4 byte)

Q44) Name the package that contains Scanner class.

Ans) java.util

Q45) Which unit of the class gets called, when the object of the class is created?

Ans) constructor

Q46) Give the output of the following:-

String n=”Computer Knowledge”;

String m=”Computer Applications”;

System.out.println(n.substring(0,8).concat(m.substring(9)));

System.out.println(n.endsWith(“e”));

]

**Ans) Background knowledge for solving the problem:**

n.substring(0,8)

the details of the substring method which is called during this is given as follows:

String substring (int beg inIndex, int endIndex)

This method returns a new string that is a substring of this string .

Now, if we focus on the n string:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Characters  Of n string | C | o | m | p | u | t | e | r | ‘ ‘  (space) |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

These are the corresponding characters of indices 0-8 of n string

m.substring(9)

String substring (int beg inIndex)

This method returns a new string that is a substring of this string . i.e. this method returns a substring whose starting point is the index which is sent as argument (of the original string) and end index (which is not mentioned as the argument of the method) is the length in integer of the original string.

**m string**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | o | m | p | u | t | e | r | ‘ ‘ | A | p | p | l | i | c | a | t | i | o | n | s |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

So, m.substring(9) will return “Applications” . n.substring(0,8).concat(m.substring(9)) this concat method will concat two strings “Computer “ (at the end of the string there is a space) and “Applications”. (concatenation:- adding two strings). So, ans is “Computer Applications” after concatenation.

String concat(String str)

This method concatenates the specified string (str) to the end of this original string . (the string through which the method concat is called)

and n.endsWith(“e”)

the details of the method which is called here is given below:

boolean endsWith(String suffix)

This method tests if this string ends with the specified suffix string.

now, n does not end with “e”. It ends with “s” (if we consider “s” as a prefix string) ..

However, if we send prefix strings like: “plications” or “cations” or “tions” endsWith function of String class would return true.

But here, it will return false.

So, ans of the first q is) **“Computer Applications”** (“ “ is to denote that it is a string) and the ans of the second q is **false**.

Q47) Give the output of the following:

1. System.out.println(Character.isUpperCase(‘R’));
2. System.out.println(Character.toUpperCase(‘j’));

Ans) **Background knowledge for solving the problem:**

**isUpperCase** is a method of Character class (under java.lang package) and it is a static method. That’s why we don’t need any instantiation of the class (i.e. object of the class) to call this function.

Static boolean [isUpperCase(int ch)](http://www.tutorialspoint.com/java/character_isuppercase.htm)  
Determines whether the specified char value (i.e. ch) is uppercase or not. If the specified character value is uppercase then it returns true. Otherwise false.

And toUpperCase is also a static method.

Static int toUpperCase(int ch)

Converts the specified char value to the corresponding upper case value if the specified char value is lowercase. Otherwise, it does not do anything. (returns the same character value) . note that, actually the ASCII value of the corresponding character is sent as parameter and the method also returns the ASCII value of the resulting character.

So, first ans) true

Second ans) J

Q48) Analyse the following program segment and determine how many times the loop will be executed and what will be the output of the program?

int p=200;

while(true)

{

if(p<100)

break;

p=p-20;

}

System.out.println(p);

Now, this is equivalent to

int p=200;

while(true)

{

if(p<100)

{

break;

}

p=p-20;

}

System.out.println(p);

Ans) **the thinking approach to the problem:**

Now, before any execution of the while loop takes place, p’s value is 200

Now, while(true) is a form of infinite loop (i.e. it will run infinite times until the (p<100) condition is satisfied so that the break statement gets executed and the control jumps out of the loop.

Now, during the first execution, the condition of the if statement block is not satisfied and p is decremented by 20. So, after first execution the value of p is 180.

During the second execution, the condition for the if statement block is not satisfied and p is decremented by 20. Now, after the second execution p’s value is 160.

During the third execution, the condition for the if statement block is not satisfied and p is decremented by 20. So, after the third execution the value of p is 140.

During the 4th execution, the condition for the if statement block is not satisfied and p is decremented by 20. So, after the 4th execution the value of p is 120

During the 5th execution the condition of the if statement block is not satisfied and p is decremented by 20. So, after the 5th execution the value of p is 100.

During the 6th execution p’s value is 100 so the if statement block is not satisfied, and p’s value is decremented by 20. So, after the 6th execution p’s value is 80.

During the 7th execution the p’s value is 80 and therefore, the if statement block is satisfied. And control jumps out of the loop (p is not decremented this time. Because the statement p=p-20 does not get executed).

So, the ans is the loop will be executed for 7 times and the output of the program is 80 (last value of p)

Q49) What will be the output of the following code**?**

int k=5, j=9;

k+=k++-++j+k;

System.out.println("k="+k);

System.out.println("j="+j);

Ans) this is a typical question from precedence of operators and associavity.

Now, ‘+’ has higher priority than ‘-‘. And ‘++operand’ (i.e. pre increment) and ‘operand++’ (post-increment) both have higher precedence or higher priority than ‘+’

k+=k++-++j+k

is equivalent to k+=k++-(++j)+k since both pre-increment and post-increment operator have same precedence and their associativity is from right to left

now, j’s value is first incremented then evaluated (since it is pre-increment operator)

k+=k++-10+k (with j’s value as 10)

Now, it is equivalent to

k+=(k++)-10+k since among the left operators, the increment operator (post-increment operator specifically) has higher precedence.

Now, it

Now, in case of (k++), first k’s value is evaluated then it is incremented.

i.e. k+=5-10+k (with k’s value as 6 for further use)

Now, both ‘+’ (addition) and –(subtraction) has same priority and they have associativity from left to right

k+=-5+k (with k’s value as 6 for further use)

k+=-5+6

k+=1

now, this compound assignment operator adds the right side value of the assignment operator to the left side value of the assignment operator. (now, remember, the left side value of x still remain unchanged. Because, no assignment operation is performed so far) So, first addition operation is performed on 5 and 1 and then the resulting value of this operation (which is 6) is assigned to k.

So, result is k is 6

j is 10.