ARRAY:

1.what is array?

It is an user defined data structure which contains same type of data element in contigious memory places.

int a[]=new int[10];//declaration can contains 10 integer value

int []a=new int[10];//also correct

double a[]={2,4,7.8,9.6,5}//declaration with assigning value

double a[]=new double[]{2,4,7.8,9.6,5};//also correct

2)what is length?

It is a property. It finds number of elements in the array.

int a[]=new int [15]; int k=a.length; sopln(k);

o/p= 15

3)

Linear search	Binary search
i)array may or may not be sortd	i)array must be sorted
ii)slow process	ii)fast process
iii)searching through out array	iii)searching in either half
iv)sequential searching	iv)searching using key value/mid value

4)advantage of binary search over linear search?

Faster process

5)disadvantage of binary search over linear search?

In binary search array must be sorted.

6)advantage of array?

i)More than one variable possible to store under single variable name

ii)elements in contagious places, hence accessing becomes first.

7)disadvantage of array?

i)size have to specify before use

ii)only same type of element possible to store

8)why is array called subscripted variable?

Array is accessed by its subscript position, hence it is called subscripted variable.

9. find size of array using primitive data.

Primitive data	Size(in byte)	Size(in bits)	default value
byte	1	8	0
boolean	1	8	false
char	2	16	'\u0000'
short	2	16	0
int	4	32	0
float	4	32	0.0f
long	8	64	0L
double	8	64	0.0d

10. an array A contains 10 double values. Print size in bits and byte and print default value of each cell.

Size in bits 10*64=640 bites

Size in byte 10*8=80 byte

default value: 0.0d

11. syntax , size(in bits) for boolean array of 10 elements

Syntax: boolean b[]=new boolean[10];

Size in bytes: 10*1=10 byte

Size in bits:10*8=80 bits.

12)find output:

```
int a[]={3,12,6,16,9,20};
int s=0;
Sopln("length="+a.length);
Sopln(Math.ceil(Math.sqrt(a[3]))
for(int i=0;i<a.length;i++)
{if(a[i]%4==0)
s=s+a[i];
}
Sop("sum ="+s);
```

o/p

length=6

4.0

sum=48

13)

```
int a[]=new int [8];
for(int i=0;i<8;i++)
{if(i%2==0)
a[i]=a[i]+3*i;
}
```

for(int i=0;i<6;i++)

Sop(a[i]);

o/p….0 0 6 0 12 0

14)all methods of String class in java.lang package.

15)

length	**length()**
Used to find number of element in the array	**Used to find number of number of character in array**
It is property	**It is method**
int a[]=new int[10]; **Sopln(a.length)->** **10**	**String st=”Kolkata** **Sopln(st.length()** **7**

16)array ,String are composite/reference/user defined data type

Array ,String passed in method is known as pass by reference

17)Any change in array must be reflected in actual parameter

Static/class/global	**Non static**
Created only once whatever may be number of object	**Created repeatedly with every object**
Can’t use non static method or member	**Can use static method or member**

String	String Buffer
Immutable	Mutable
For every changes new object is ceated	Change made on same object

Oop special:

Encapsulation: binding or wrapping up of data and method together, abstraction possible for Encapsulation

Abstraction: hiding details, secured, implemented through encapsulation

Inheritance: subclass, superclass relation, accessing data of one class by another

Polymorphism: same external feature but different internal interface, method overloading(static binding) method overriding(dynamic binding)

Class: blue print of object, object factory

Object: unique entity of class,

Super class/base class/parent class(from where data member inherited)

Sub class/child class/derived class(which inherits)

ACCESS SPECIFIER:

private can be accessed by that class only and cannot be inherited

default/friendly can be accessed by all class of that package

protected accessed by all classes of that package and first subclasss in other package

public accessed by all classes of that package as well as other package

Static :

i)not bounded to object,

ii) bounded to class,

iii)change in any object effect on other

iv)static method can not use non static for that variable at a time

v)all object of a class contains same value

Non static/ instance variable

i)bounded to object

ii)variables may have different values for different object of the same class

Local Variable: Scope only the method block where they are declared

Searching	sorting
i)search an element in the array	i)arrange all elements in the array in proper order
ii)two type search: linear , binary	ii)two type :bubble sort, selection sort

Conditional statement or if-else

1.What is sequential statement?

This type of statements are being executed sequentially. It requires default flow of statement.

2.what selection or conditional statement?

It allows to choose a set of instruction for execution depending upon expressions true or false.

e.g if-else, switch

3.What is compound statement?

One or more than one statements under a pair of curly bracket is known as compound statement.

e.g

```
if(a>b){

s=s+a+b;

sop(s);

}
```

3.difference between if else switch

If-else/if/if-else-if	**switch**
It can check any relational expression	It can check only equality
It can perform test on floating point also	It cannot work on floating point
If-else	**switch**
It is bi-directional	It is multiple branching statement
if	**switch**
It is unidirectional	It is multiple branching statement
If-else-if	**switch**
It is slower	It is faster

4. What is fall through?

If 'break' is not applied after each case in switch statement then all cases will be executed until it gets break. It is called fall through.

e.g switch(ch)

{case 1:

Sop("one");

case 2:

Sop("two");

break;

case 3:

sop(“three”);

break;

default:

sop(“invalid”);

}

I/P ch=1..O/P one two

I/Pch=2… O/P two

5.what is default statement?

In a switch case if no other case constant matches with switch value then default is executed.

e.gswitch(ch)

{case 1:

Sop(“one”);

break;

case 2:

Sop(“two”);

break;

case 3:

sop(“three”);

break;

default:

sop(“invalid”);

}

I/P ch=5 O/P invalid

6.What is use of ‘break’ statement in java

‘break’ is used to avoid fall through in a switch case. Otherwise all cases will be executed from matching case until it gets break.

e.g switch(ch)

{case 1:

Sop(“one”);

case 2:

Sop(“two”);

break;

case 3:

sop(“three”);

break;

default:

sop(“invalid”);

}

I/P ch=1..O/P one two

I/Pch=2… O/P two

7. What are different type of case constant?

char, int

8.What is error? What are different kind of error?

Any problem occurs at compile time or runtime of a java program is known as error. Three type of error.

i)Syntax error/compile error- This error due to grammatical mistake at compile time.

e.g missing semicolon, bracket mismatch etc

ii)logical error-> This is due to error in program logic development. Output must comes but not desired.

e.g Incorrect formula.

iii)Runtime error: This occurs during execution of a java program due to violation of java rules and limitation in java execution environment. e.g Division by zero.

9.What is scope of variable?

It indicates the program region through which a variable is accessible in a program.

e.g void sum(int a, int b) {

int s=0;

for(int i=1;i<=10;i++)

{s=s+i;

}

}

Here scope of s,a,b are through sum() method and scope of ‘i’ through loop

Write down name of following error:

i)semicolon not given at the end of statement:

ii)addition operation in exchange of multiplication

iii(switch(a>2)

iv)sop(56/0)

v)int a=5;b=7;

vi)square root of negative number:

answer:

i)syntax error

ii)logical error

iii)syntax

iv)runtime

v)syntax

vi)runtime

1. What is constructor? Or characteristics of constructor.

It is a member method of a class having same name as that of the class. It has no return type.

It initializes data member at the time of creation of object and it is called automatically at the time of creation of object.

2. What are different kind of constructor?

Two type.

i)Non parameterized constructor: It has no parameter. It initializes data member by default value.

ii)Parameterized constructor: It has parameter and it initializes data member by parameterized value

3. What is constructor overloading? Give example.

When a class contains more than one constructor with different parameter list then they are said to be overloaded.

```
Class Employee
{String name;
double sal;
int age;
Employee()//non parameterized constructor
{name="";
Sal=0.0d;
age=0;
}
Employee (String n, double s, int a)//parameterized constructor
{name=n;
Sal=s;
age=a;
```

}

}

constructor	Method
i)It has same name as that of the class	i)It has any valid name
ii)It has no return type	ii)it must has a return type
Iii)It initializes data member	iii)It has any purpose
iv)It is implicitely called	iv)It is explicitely called

4.difference between constructor and method

5.what is default constructor?

If no other constructor provided by programmer then the constructor provided by compiler is known as default constructor. Here data member initized by default value

Myclass ob=new Myclass();//default constructor

6.

User defined data/composite data/nonprimitive data	Primitive data
They are dependent on primitive data	They are independent
Length depends on user	Length is fixed
They are identified by particular block where they are defined	They are identified by through out the program
e.g..class, array	e.g int,long,float etc

What is this keyword?

i)It removes name confliction

ii)It refers current object

class Abc

{int a;

```
Abc(int a)

{this.a=a;

}

}
```

1. **Call by value**	**Call by reference**
i)Here method is called using primitive data	i)here method is called using reference data
ii)Any change in formal parameter does not reflect actual parameter	ii)Any change in formal parameter reflects the actual parameter
iii)Here value is copied from actual to formal parameter	iii)here address is copied from actual to formal parameter

e.g void calc(int x,int y) {
x++;y++;
}
	e.g void calc(Myclass ob)
	{ ob.x++;
	Ob.y++;
	}

2. **Actual Parameter**	**Formal Parameter**
i)It is used in method calling statement	i)It is used in method signature
ii)Data type are used with formal parameter	ii)Data type are used for call side parameter
iii)It takes value from outside	iii)It takes value from actual parameter

e.g
void main(){
int a=5,b=2;
sum(a,b);//actual parameter/argument/method calling statement
}
	e.g void sum(int x,int y){->formal parameter/parameter/method signature/method definition
	
	……
	}

Impure method	**Pure method**
i)It changes state of object	ii)It does not change the state of object
ii)It may or may not return value	ii)It returns a value

e.g void setdata(int x){
this.x=x;
}
	e.g int getdata(){
	return x;
	}

Static method	**Non static method**
i)It is not associated with object	i)It is associated with object
ii)It can use only static variables or method	ii)It can use static as well as non static variable and method
e.g	e.g
Static variable/global/class variale	**Non static variable/instance variable**
i)It is not associated with object	i)It is associated with object

ii)It is used by static as well as non static method	ii)It can be used by non static method only
e.g	e.g
Local variable	**Instance variable**
i)It is not associated with object	i)It is associated with object
ii)It can be accessed by the block where it is declared	ii)It can be accessed by all methods of the class

e.g class abc{
int x;
void sum(int a){->'a' local variable
…..}
}
	Class abc{
	Int x;//->'x' instance variable
	Void set(){…}
	}

Local variable	**Global variable/static/class**
It is accessed by the block where it is declared	It is accessed by all methods of the class
It is stored in stack memory	It is stored in heap memory

e.g class abc{
static int x;
void sum(int a){->'a' local variable
…..}
}
	Class abc{
	static int x;//->'x' global variable
	Void set(){…}
	}

Access specifier: It indicates the program region through which a variable or method is accessiable.

Four type:

i)private: Accessed by that class only. It is preceded by private keyword. It is most restricted.

ii)default: Accessed by all class of that package. No keyword is used here

iii)protected: Accessed by all class of that package and next subclass of other package. It is preceded by protected keyword

iv)public: Accessed by all classes of that package as well as all classes of other package. It is preceded by public keyword. It is least restricted

What are different way of invoking method?

Two.

i)call by value/pass by value

ii)call by reference /pass by reference

what does return statement do?

i)It returns value to calling method

ii)It returns control to calling method

What do u mean by void?

'void' means the method returns no value

What is method?

It is sequence of some declarative and executable statement. It is also known as subroutine or sub program and is called from different part of program

Functionality of method.

i)To cope with complexity- It breaks a program into sub program to cope with complexity

ii)Hiding details- private members are used from methods ,which are not accessible from outside

iii)Reusability: Once it is written can be used from different part

What is method overloading?

When a class contains more than one method having same name but different parameter list then the methods are said to be overloaded

e.g void sum(int a, int b)

{……

}

Void sum(double a,int b,int c)

{….

}

Void sum()

{…..

}

What is method prototype?

It is the first line of a method which contains return type , method name and parameter list.

Int sum(int a,int b,double c)-> method prototype

What is method signature?

The parameter list (number of parameter and data type of parameter) in a method is known as method signature.

Int sum(int a,int b, double c);

Here method signature ..int a,int b,double c

e.g of Static variable: and nonstatic /instance variable and static, non static method

class abc

{

int a;//instance/non static variable variable

static int b;//static variable

void change()

```
{a++;b++;// non static method can use both static and non static variable
}
static void change1(){
b++;//static method can only use static variable
}
}
```

History Development of java(chapter 2)

1.Feature of Bluej

i)It is windows based

ii)It facilitates us with sample program

iii)Debuging and correcting error is easily done

iv)Both compiled and interpreted

2.what is comment line? What are different kind of comment line?Give example

Comment lines are used to make the code understandable to other . It is not compiled.

There are three type of comment line

i)Single Line comment(//)..when programmer needs to comment on single line

ii)Multiline comment(/*……*/)when programmer needs to comment more than one line

iii)Documenting comment(/…..*/)when programmer needs to include some text document which is some description about the program**

3.what is byte code?

After java compilation an intermediate code is created called byte code. It is platform independent. It makes java platform independent

4.what is JVM?

It is java virtual machine. It is an interpreter. It helps to run byte code from machine to machine.

5. what are different kind of java program?

Two type of java program.

i)java stand alone program(It is used for personal purpose in personal machine)

ii)java applet program(it is used for web browser)

6.what is package? What is default package in java?

A package is a group of co related classes which is included in a program by import keyword such that the user can use its implicite facility in a program.

e.g java.util, java.io,java.lang

default package in java java.lang

7.what is import keyword?

'import' is keyword which is used to include a package with its classes in a program.

e.g import java.util.*; import java.io.*;

8)what is java API?

Java Application Programming Interface. It contains all prewritten packages, classes, methods etc.

e.g javaAPI package java.util, java.io

javaAPI class Scanner, Math

java API method Math.pow, Math.sqrt

9)what is final keyword?

'final', is a keyword which makes a variable constant through out the program.

e.g final int a=10;

10)what is 'new' keyword?

'new' is a keyword which is used to create memory space dynamically for an object.

e.g Scanner sc=new Scanner(System.in);

11)what is Math function?

All function helps in mathematical calculation is known as Math function. These all function are of java.Math class

e.g Math.sqrt(), Math.pow etc

12. difference between compiler and interpreter

compiler	interpreter
Converts whole source code into object code at a time	Converts whole source code into object code line by line
It is faster	It is slower
Displays error of whole program together	Displays error of whole program line by line

13.difference between sop() and sopln()

System.out.print()	System.out.println()
i)after printing cursor remains on the same line	i)after printing cursor moves to nextline
ii)it must contain parameter	It may or may not contain parameter

14.difference between souce code and object code

Source code	Object code
i)it is written by programmer	i)It is converted from source code by compiler

ii)it is human readable	ii)it is machine readable
iii)it is high level language	iii)it is low level language

15.difference between normal compilation and java compilation

Normal compilation	Java compilation
i)here source code is compiled only	i)Here source code is compiled as well as interpreted
ii)it creates native code	ii)it creates byte code
e.g c,c++	e.g java.python

16.difference between next() and nextLine()

next()	nextLine()
i)it takes input till space	i)it takes input including space
ii)after taking input cursor remains on the same line	ii)after taking input cursor moves to nextLine.

What is Library package? Give example.

The packages already written in java compiler is known as library package.

E.g java.util, java.io etc

What is Library class? Give example.

The classes already written in java compiler is known as library class.

E.g Scanner ,Math etc

What is Library method? Give example.

The methods already written in java compiler is known as library method.

E.g Math.pow(), nextInt() etc

Four type of access of specifier

i)private:accessed by that class only

ii)default :accessed by all classes of that package

iii)protected : accessed by all classes of that package and next subclass in other package

iv)public: accessed by all classes of that package and all classes of other package

2)i)new operator: creating memory space for object dynamically

ii)dot(.)operator: it is used to call package, classes, methods etc.

3)static variable: created only once what ever may be number of object .

All object commonly share the static variable

It is related with class not associated with object

4)non static variable: It is associated with object

It is created separately for all object

5)static method: it is related with class . not associated with object.

It can use only static variable and method

6)non static method: it is associated with object

It can use both static and non static member and method

Oop feature:

i)encapsulation (wrapping up of data member and member method)

ii)abstraction(hiding data)

iii)inheritance: capability of one class to inherit data member and member method of another class

iv)polymorphism: it is method overloading and method overriding(same external feature and different internal interface)

class: blue print of object, object fctory

object: instance of class

```
class ABC
{int a;
static int b;

void input()
{...input
}
```

```
void main()
{
ABC ob=new ABC();
ABC ob1=new ABC();
ob.input();
ob1.input();
}
}
```

Four type of access of specifier

i)private:accessed by that class only

ii)default :accessed by all classes of that package

iii)protected : accessed by all classes of that package and next subclass in other package

iv)public: accessed by all classes of that package and all classes of other package

2)i)new operator: creating memory space for object dynamically

ii)dot(.)operator: it is used to call package, classes, methods etc.

3)static variable: created only once what ever may be number of object .

All object commonly share the static variable

It is related with class not associated with object

4)non static variable: It is associated with object

It is created separately for all object

5)static method: it is related with class . not associated with object.

It can use only static variable and method

6)non static method: it is associated with object

It can use both static and non static member and method

Oop feature:

i)encapsulation (wrapping up of data member and member method)

ii)abstraction(hiding data)

iii)inheritance: capability of one class to inherit data member and member method of another class

iv)polymorphism: it is method overloading and method overriding(same external feature and different internal interface)

class: blue print of object, object fctory

object: instance of class

```
class ABC
{int a;
static int b;

void input()
{...input
}
```

```
void main()
{
ABC ob=new ABC();
ABC ob1=new ABC();
ob.input();
ob1.input();
}
}
```

JAVA THEORY

Q. What is a class?

Ans: A class is a blueprint or prototype of a group of objects having similar characteristic and behaviour.

Q. What is object?

Ans: An object is an instance of a class. It possesses all the characteristics and behaviour as defined in a class.

Q. What are the characteristics / features of Java?

Ans: Java is a case sensitive language.

It is Platform independent language.

It is an object oriented language.

It is a real-life oriented language.

It uses re-usability.

It uses readability

Java is also a compiler language.

Java is a structured language.

Q. Define Object Oriented Programming (OOP).

Ans: An Object Oriented Programming (OOP) is a modular approach which allows the data to be applied within stipulated program area. It is also provided the reusability feature to develop productive logic which means to give more emphasis on data. It has been developed to increase the programmer's productivity and also to overcome the weaknesses of traditional approach of programming.

Q. Name the basic elements of OOP (Principles of OOP). / Name some OOP principles.

Ans: The basic elements of the Object oriented programming (OOP) are: Class, Object, Data Abstraction, Encapsulation, Inheritance, Polymorphism, Data Hiding.

Q. Concept of objects and class.

Ans: A class and an object have a close relationship in Object Oriented Programming. Class is used to create various objects which possess same characteristics and behaviour within the class. So we can say that class is a blue print or prototype of an object. The characteristics and behaviour of the class is declared at the time of its creation. Each object of the class possesses same features defined within the class. e.g. If Car is the class with the characteristics / make (name of the company), colour, model and version, then a car, Maruti which is white in colour, Swift DZire, diesel version is used as an object of the class 'Car'.

Q. What is Abstraction?

Ans: Abstraction is the act of representing data members of a class without knowing the background details.

Q. What do you mean by Encapsulation?

Ans: Encapsulation is a feature of JAVA by which we wrap-up data and functions, into one single unit known as CLASS.

Q. Define Polymorphism.

Ans: Polymorphism is a feature of Java, by which one object can behave differently under different circumstances. It is implemented in java using function overloading.

Q. Define Inheritance.

Ans: Inheritance is a feature of Java by which one class can acquire the properties of another class. The class whose properties are acquired are known as base class / super class and the class which acquires the properties of the base class is known as derived or sub-class.

Q. What is a class and an object with reference to java?

Ans: A class is a blueprint or prototype of a group of objects having similar characteristic and behaviour.

An object is an instance of a class. It possesses all the characteristics and behaviour as defined in a class.

Q. Why class is called an object factory?

Ans: A class is called an object factory as it is a template, using which any no of objects can be created.

Q. Why is a class known as composite datatype?

Ans: A composite data type is one which is composed with various primitive data type. A class defined with various primitive data types such as int, double etc, so it is known as composite data type.

Q. Why is an object also known as an instance of a class?

Ans: Since, an object possesses instant variables and member method defined within the class. This is the reason why an object is called an instance of a class.

Q. Define Byte code.

Ans: It is a set of pseudo language instructions which are directly interpreted by the JVM thereby making java a platform independent language.

Q. Define the term Platform independence.

Ans: A java program is compiled into a byte code. A byte code is a pseudo-machine language instruction, which is directly interpreted by the JVM (Java Virtual Machine), thereby making Java a platform independent language. Platform independence program can run identically on Windows and Linux.

Q. Explain "Java is a Case Sensitive language".

Ans: Java is a case sensitive language as the upper case and lower case letters are distinguished by the language. It treats upper case variable and lower case variable differently.

Q. What do you mean by Structure of a Java program?

Ans: A java program is a block / blocks inside a block.

Q. Define Block.

Ans: A Block is a group of statement or function enclosed between { }. A compound statement is also called a block.

Q. Define JVM.

Ans: JVM - Java virtual Machine

Q. Define SCOPE.

Ans: The SCOPE of a variable is the range of validity of a variable. A variable is valid only in the Block, in which it is declared.

Q. Define Tokens.

Ans: A token is the smallest individual unit of a program that is meaningful to the java compiler.

Q. What are the different types of tokens?

- **Identifiers**
- **Literals**
- **Separators -, &, ;**
- **Operators +, -, ++, >=, <=**
- **Keywords**

Q. Define Identifiers.

Ans: Identifiers: are names which are defined by the programmer in the program.

In java we have 3 identifiers -

1. **Class name**
2. **Function / Method name**
3. **Variable name**

Q. What are the rules of identifiers in Java?

Ans: It should start with an alphabet ONLY.

It can be any length (upto 256 characters)

Dollar and underscore are the only two symbols allowed in Java (anywhere after the first character)

Q. Define Literals. Explain each literals with example.

Ans: The constant used in a java programs which remain fixed throughout the program is called literals or constants.

Integer Literals: The numbers which are represented without decimal point are called Integer Literals. These are the whole numbers having positive or negative values. e.g. 14, 5

Real Literals: Real Literals are also called floating point constants. They represent numbers with decimal point. e.g. 24.6, 0.007.

Character Literals: The constants, which are alphanumeric in nature, are called character literals. All alphabets upper or lower case, digits special characters can be terms as character literals. e.g. 'A', 'd', '3', '*' etc

String Literals: String is a set of alphanumeric characters. A group of characters enclosed within a pair of opening and closing double quotes is known as string literals. e.g. "Ajay", "games"

Q. What do you mean by an operator? Give two examples of mathematical operator.

Ans: Operators are special symbols which perform a pre-defined task on a set of operands. + and * are two mathematical operators.

TYPES OF OPERATORS:

1) **ARITHMETICAL OPERATORS**
2) **RELATIONAL OPERATORS**
3) **LOGICAL OPERATORS**

Arithmetical Operators: The operators, which are applied to perform arithmetical calculations in a program are known as arithmetical operators.

Types of arithmetical operator,

- **Unary operators (++, --): An arithmetical operator, which is applied with a single operand. Unary Increment Operators (++) increases the value of an operand by one whereas Unary Decrement Operators (--) decreases the value of an operand by one.**

  PREFIX: The value of the variable first changes and then the changed value is used to evaluate the expression. Eg: ++a (change and use)

  POSTFIX: The existing value of the variable is used to evaluate the expression & then the value of the variable changes. Eg: b++ (Use and change)

- **Binary operators (+, -, *, /, %): An arithmetic operator, which deals with two operands is known as Binary Arithmetic Operator.**

ARITHMETICAL OPERATORS	SYMBOLS	
1. Multiply...	*	Same Preference – Left to Right
2. Divide	/	
3. REMAINDER OR MODULUS	%	
4. Add	+	Same Preference – Left to Right
5. Subtract	-	

RELATIONAL OPERATORS: These operators are used to show the relationship among the operands. Relational operators compare the values of the variables and result in 'True' or 'False".

The different types of Relational Operators are as follows:

RELATIONAL OPERATORS	SYMBOLS
1. Greater than	>
2. Greater than equal to	>=
3. Less than	<
4. Less than equal to	<=
5. Equal to	==
6. Not equal to	!=

LOGICAL OPERATORS: Java uses logical operators AND (&&), OR (||) or NOT (!). Logical operators are used when we want to check multiple conditions together. We can combine many relational express using Logical and operations. The result will be a boolean type.

LOGICAL OPERATORS	SYMBOLS
1. NOT	!
2. AND	&&
3. OR	||

Logical AND (&&): The AND operator results in true if both the expression (comprising its operands) are true.

5>3 && 3<5, Results in true because both the expressions are true.

6==6 && 3>0, Results in true because both the expressions are true.

5! =5 && 4==4, Results in false as the first expression is false.

Logical OR (||): The OR operator is used to combine two conditional expressions. It will result in true if either of the two conditions (expression) is true otherwise false.

5>4 || 8>12, Results in true because 5>4 true.

3>7 || 5<=4, Results in false because both the expressions are false

2<0 || 2<12, It will result in true because second expression is true.

Logical NOT (!): Logical NOT operator is applied when you want to revert the outcome of an expression. It is a unary operator because it uses a single operand.

!(8>3), False, because 8>3 is true

!(5<7), False, as 5<7 is true

!(3<0), True as 3<0 is false

JAVA SHORTHAND ASSIGNMENT OPERATORS

x+=10	**equivalent to**	**x=x+10**
x-=10	**equivalent to**	**x=x-10**
x*=3	**equivalent to**	**x=x*3**
x/=2	**equivalent to**	**x=x/2**
x%=z	**equivalent to**	**x=x%z**

Q. Define Precedence of operators.

Ans: The order in which the operators are evaluated in an expression is known as precedence of operators. e.g. ++, %, >=, &&

Q. What is a ternary operator? Give an example.

Ans: The conditional operator is also known as the ternary operator because it operates on three operands.

Syntax: condition ? true value : false value

Example 1: int x = (y==24) ? 400 : 200;

Output : Here if the value of variable y is 24, then variable x stores 400 otherwise 200.

Q. **Why is the conditional operator also known as the ternary operator?**

Ans: **The conditional operator is also known as the ternary operator because it operates on three operands.**

Syntax: **condition ? true value : false value**

Q. **Define Keywords. Name any two keywords in Java.**

Ans: **Keywords are words which are predefined for the java compiler and have a special meaning for the java compiler. public, static, void, import**

Q. **What is the role of the new keyword?**

Ans: **The keyword new is used to allocate space in the dynamic memory for the storage of data and functions belonging to an object in java programming. It creates a new instance of a class.**

Q. **What is the role of dot operator?**

Ans: **The dot operator facilitates invoking members of the class to carry out the tasks. Example : java.util.*;**

Q. **What is a variable? Give two examples.**

Ans: **A Variable is a named memory location which stores a value of a specified data type. Example: int a, double b.**

Q. **What is an expression? Give an example.**

Ans: **An expression is a combination of operators and operands which returns back a value. Ex: a+b**

Q. **Define Statement.**

Ans: **If an expression is assigned to a variable, it is known as statement.**

c=a+b;

Q. **What is a compound statement?**

Ans: **Compound statements are statements that contain lists of statements enclosed in braces { }**

Example:

```
{
        statement;
        statement;
}
```

Q. What are the two types of data in java?

Ans: There are mainly two types of data:

Primitive data are pre-defined or inbuilt in the java compiler

Reference data are defined by the programmer with reference to other data types.

Types of Data

i) Primitive		**ii) Reference**	
1	**char – 1 character 'A', '8', '!'**	**1**	**classes**
2	**byte – integers**	**2**	**objects**
3	**short – integers**	**3**	**arrays**
4	**int – whole number**	**4**	**interfaces**
5	**long – ten digit numbers**		
6	**float – real numbers v, decimal numbers**		
7	**double – real numbers, decimal numbers**		
8	**boolean – true / false**		

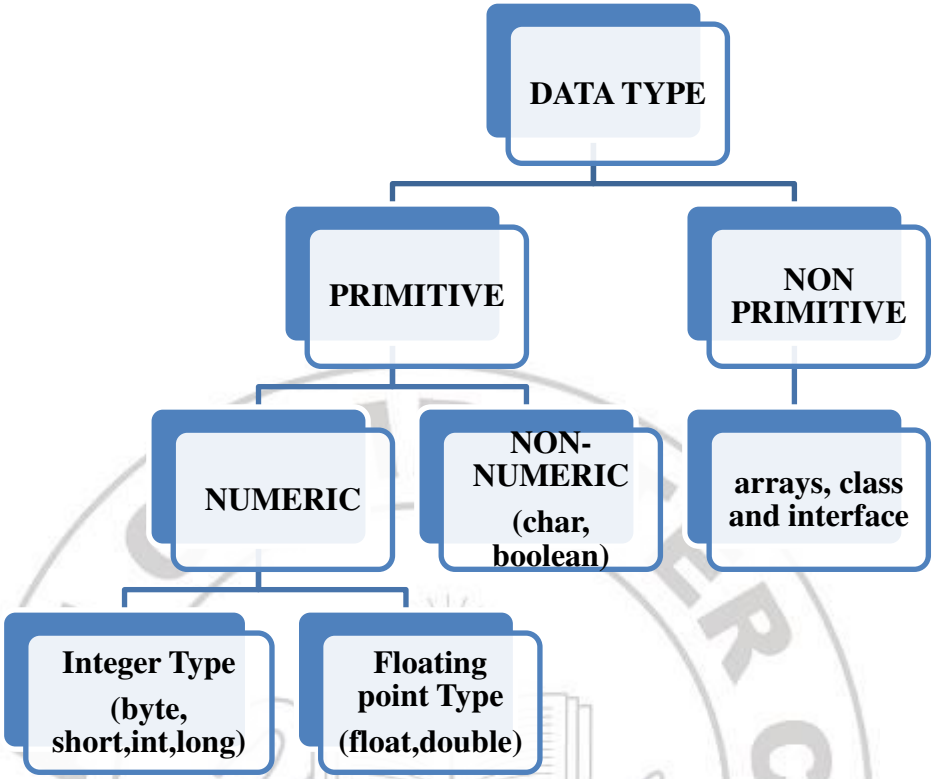
Q. Define Data Type.

Ans: The data type represents that the particular entity is used for which kind of operation.

The data types are divided into two categories:

1) **Primitive or pre-defined data types: These data types are available with Java language. They are transferred in the memory as soon as Java program is loaded in the computer system. The primitive data types supported by Java are: byte, short, int, long, float, double, char, boolean.**

2) **Non-Primitive or Derived or Reference data types: These data types are formed with the help of primitive data types. The non-primitive data types supported by java are: arrays, class and interface.**
A non-primitive data type or reference data type is used to store the memory address of an object.



Q. What are the default values of int and float?

Ans:

Data type	Size	Range of values	Default Value
char	2 bytes		'\u0000'
byte	1 byte	-2^7 to 2^7-1(-128 to 127)	0
short	2 bytes	-2^{15} to $2^{15}-1$(-32768 to 32767)	0
int	4 bytes	-2^{31} to $2^{31}-1$	0
long	8 bytes	-2^{63} to $2^{63}-1$	0 l
float	4 bytes	-3.4×10^{-38} to 3.4×10^{38}	0.0f
double	8 bytes	$-1.7 \times e^{-308}$ to $1.7 \times e^{308}$	0.0d
boolean	1 bit		false

8 bits=1byte
$e^2 = 10^2$

$e^{-2} = 10^{-2}$

Q. What are the values of a Boolean data type?

Ans: Boolean data type either stores true or false values.

Q. Name 2 decision making statements in java.

Ans: Two decision making statements in java are

a) if – else if b) switch case

Q. What is the function of the import statement?

Ans: The import statement helps us to use or include classes of an external package.

import package.class

Q. What is the role of default in a switch statement?

Ans: The default statement is executed if the switch statement does not match any of the given case. In the absence of the default statement in a switch no job is performed if the switch variable does not match any of the given cases.

Q. What happens if a default is not included in a switch statement?

Ans: If default is not included in a switch statement and if the user is inputting any such value which does not match with any of the given case then no job is performed.

Q. What is the purpose of break in a switch statement?

Ans: The break statement skips the subsequent statements in a loop / switch and transfers the control out of the loop / switch block.

Q. What do you understand by fall through with respect to switch case?

Ans: The switch statement has a fall through property by which it keeps executing all the subsequent cases till it completes executing the entire switch block. To prevent this a break statement is given after the case thereby transferring the control out of the switch block on the completion of desired case.

The default statement is executed if the switch statement does not match any of the given case. In the absence of the default statement in a switch no job is performed if the switch variable does not match any of the given cases.

Q. Define multiple branching statement in java.

Ans: The multiple branching statement in Java is switch. Switch statement works on the basis that the value of a variable is.

Q. Is it compulsory to use a default in a switch? What will happen if default statement is not given?

Ans: It is not compulsory to use a default in a switch. In the absence of a default statement, no job is performed if the switch variable does not match any of the given cases.

Q. Name any 3 jump statement in java. Explain each of them.

Ans: Java provides three jumps statements:

1. **Return: It is a keyword and this statement returns back the program control from a called function to its calling functions.**
2. **Break: Break statement terminates the control from the current loop. This statement is appear inside the loop and switch case statement. This statement skips the rest of the loop and jumps over the statement following the loop and switch case.**
3. **Continue: This statement abandon the current iteration of the loop by skipping over the rest of the statements in the loop body and immediately transfer the control to the evolution of the test expression of the loop for the next iteration of the loop.**

Exception Handling

Q. Define Exception.

Ans: Exception: An exception is an anomalous situation / error which may occur during the execution of the program / runtime.

Q. What is an exception handler?

Ans: Exception Handler: An exception handler is a class which contains the definition of all possible anomalous situation that may arise during runtime, and the cause of action to be taken there upon. The job of the exception handler is not only to ensure the smooth functioning of the program but also the smooth execution of it. eg. Example - throws, try, catch.

Q. Name 2 Exception handlers.

Ans: Try and catch are two exception handlers: The normal processing sequence to be performed is included in the try block. The sequence of instruction to be performed on the occurrence of an exception included in the catch block

Q. What do you mean by type casting?

Ans: Type casting is the forced conversion of one primitive data type into another using the Type caste operator (). The type caste operator is () with the data type to be converted into, being specified within ().

Q. What is type conversion? Explain the two types of type conversion.

Ans: The process of converting one primitive data type into another is called Type Conversion. Java facilitates the type conversion in two forms:

i. Implicit type conversion: An implicit type conversion is a conversion performed by the compiler automatic without programmer's intervention. An implicit conversion is applied generally whenever differing data types are intermixed in an expression (mixed mode expression), so as not to lose information.
e.g. int a; double b; double c;
c=a+b;

ii. Explicit type conversion: An explicit type conversion is user-defined that forced an expression to be of specific type with programmer's intervention.
e.g.: int a; int b; double c;
b= (int) (a*c);

Loop

Q. What is loop? Name any one.

Ans: Loop – A loop is an iteration statement, which repeats a job a required number of times.

```
for (initial expression; test expression; update expression).

for (int i=1; i<=5; i++)

{

        System.out.println (i);

}
```

Q. What is the infinite loop? Give an example.

Ans: An infinite loop is a loop which never terminates.

EX-1

```
for(;;)
{
        System.out.println ("LMB");
}
```

EX-2

```
while (true)
{
        System.out.println ("LMB");
}
```

Q. Define Dummy Loop.

Ans: If we use semicolon (;) after for loop it means for loop does not have its body and it has no statement inside the loop. The loop will initialize and terminate within its bracket. This type of loop is called DUMMY LOOP. eg.- for (int i=1; i<=10; i++);

Q. Define Conditional loops

Ans: Conditional loops repeat a given job as long as a specified condition is true.

Q. What is the role of do in a do-while loop?

Ans: The 'do' statement marks the beginning of the do while loop. It executes the block of codes at least once in the do-while loop without checking any condition.

Q. What is the use of a for loop? Explain the for loop with an example.

Ans: The for loop is used for fixed iterations in the program.

Syntax- for (initial expression; test expression; update expression)

```
{
    statements;
}
```

Example

```
class test
{
    public void(int n)
    {
            int a, sum;
            for (a=1; a<=n; a++)
            {
                System.out.println (a);
                sum = sum+a;
            }
    }
}
```

Q. Explain the while loop with an example.

Ans: while (): This loop is used to process one or more than one statements till the conditions is true/satisfies and the loop terminates as soon as the condition becomes false.

The while loop is also known as Entry Controlled Loop because before entering into the loop first the condition is checked and if it is true, the control enters into the loop and executed all the statements given within curly braces of while, if the entry conditions is false, the loop terminates and executes those instructions given out of the while loop.

Syntax:

```
while (test/ending condition)
{
statement;
statement;
-------
Increment/decrement statement or update statement
}
```

Q. Write the syntax and one example of if-else.

Ans: Eg. if (a= =10)

```
        System.out.println ("Executes")
else
        System.out.println ("Not executes")
```

A Conditional Statement in java is 'if'

Syntax

```
if (condition)
true; ob
else
false; ob
```

Q. Define Static and Non-static Method.

Ans: Static – In order to access a class member declared as static, an instance of the class need not be created, as the member is directly accessible to the class.

Non-static:- In order to access a non-static class member, an instance of the class has to be created, as the member is accessible only through the instance.

Q. Write a Java statement to create an object lamartiniere of the class School.

Ans: School lamartiniere = new School ( );

Q. Write the assignment statement for x+=2.

Ans: x=x+2;

LIBRARY CLASS

Q. Define Library class.

Ans: Library class – Library classes are classes which are predefined or inbuilt in JAVA.

Data InputStream

BufferedReader & InputStreamReader

Scanner

Q. Define package

Ans: Package – A package is a collection of related classes and interfaces.

Creating Objects.

Syntax:- classname objectname = new classname

The word new creates a new instance of a class

Relation of BufferedReader & InputStreamReader:_

The object of InputStreamReader is taken as the parameter of BufferedReader.

InputStreamReader read = new InputStreamReader (System.in)

BufferedReader br = new BufferedReader (read)

or

BufferedReader br=new BufferedReader (new InputStreamReader (System.in);

Q. Name the Java keyword that:

a) Converts a String to primitive int datatype.
b) Creates a new instance of a class.

Ans: a) Integer.parseInt(s)

b) classname objectname=new classname();

Q. Name the keywords that:

a) returns the control to the calling function.
b) Allocates memory space for an object.

Ans: a) return

b) new

Q. Define Temporary object.

Ans: A temporary object is an object which gets created to perform its job and then ceases to exist.

Q. What is the function of readLine?

Ans: ReadLine: - It accepts the value from the user in string form.

Integer.parseInt ():- It converts a String to int.

For

int :- Integer.parseInt();

double :- Double.parseDouble();

float :- Float.parseFloat();

String :- br.readLine();

Wrapper Classes

Q. Define Wrapper class..

Ans: Wrapper classes are classes which wrap up a primitive data type.

OR

Wrapper classes are classes which help us to create objects of a primitive data type.

Data Type	Wrapper Class
byte	Byte
char	Character
short	Short
int	Integer
float	Float
double	Double
long	Long
boolean	Boolean

char x= (char)br.read(); statement

char x= sc.nextLine().charAt(0); function

Q. **Write the statement to input a character in java.**

Ans: **char x=sc.next().charAt(0);**

Q. **Write a statement to convert string into int.**

Ans: **int a=Integer.parstInt(str)**

Q. **Write an input statement for float.**

Ans: **float b=Float.parseFloat(str).**

Q. **How many types of errors are there?**

Ans: **There are three types of errors.**

Syntax / compiled time error

Runtime error

Logical error

Q. **Define Access Specifiers.**

Ans: **Access specifiers are keywords which specify the accessibility of a class. i.e places from where a member class is accessible.**

Q. **Define all access Specifiers.**

Ans: **PUBLIC : Public is an access specifier which can be accessed by the classes and sub classes of the same package as well as the classes and the sub classes of the other packages**

PRIVATE: Private is an access specifier which can be accessed only by the classes in which it is created and neither by the sub classes of the same package nor by the classes and the sub classes of the other packages

PROTECTED: Protected is an access specifier which can be accessed by the classes and sub classes of the same package but not by the classes and the sub classes of the other packages

	SAME PACKAGE		OTHER PACKAGE	
	CLASSES	SUB-CLASSES	CLASSES	SUB-CLASSES
PUBLIC	YES	YES	YES	YES
PRIVATE	YES	X	X	X
PROTECTED	YES	YES	X	X

FUNCTION

Q. What are the arguments of a method? How many arguments can a method contain?

Ans: The argument of a function specify the number and type of values to be passed to a function in order to perform its job. The argument of a method is a comma-separated list of variables of a method referred to as its arguments or parameters. A methods / functions may be without any parameters, in which case the parameter list is empty.

Q. Define function prototype statement?

Ans: The function prototype statement is the first line of the function, which denotes the return type of the function, function name and the number and types of parameters to the function.

void	**area**	**int a, int b**
return type	**name**	**parameters / arguments**

Q. Why do we use void in function prototype statement?

Ans: The void keyword is given to a function prototype statement to meet the demand of the syntax. The void keyword signifies that the function does not return back any value to the calling function.

Q. What is the role of the void keyword in a function prototype statement?

Ans: The void keyword signifies that the function does not return back any value to the calling function.

Q. How many values can a function return?

Ans: A void type function does not return any value whereas non-void type / with return type function returns only 1 value.

Q. What are the uses / advantages of using functions?

Ans: The followings are the uses / advantages of functions:-

a) **Functions reduces complexity of code.**
b) **Functions gives program a better structure.**
c) **Functions enhances re-usability of code.**
d) **Functions hides implementation details.**

Q. Write the function prototype for the function "sum" that takes an integer variable (x) as its argument and returns a value of float datatype.

Ans: float sum(int x)

Q. What do you mean by invoking a function? What are the ways of invoking functions in java?

Ans: A process of using a method in a program is referred as calling a method or invoking a method. Call by value and call by reference are the two ways of invoking functions.

Q. Define Function Overloading.

Ans: Function overloading is a situation in which a class has more than one function having the same name but different parameter list / argument list / signature.

Q. Which OOP principle implements function overloading?[2007]

Ans: Polymorphism implements function overloading.

Q. Define instance variable.

Ans: An instance variable which is present in an instance of the class. i.e. an object.

Function of pow()

The pow function returns the value of a no. raised to the power of another no.

eg. double c=Math.pow(2,3)

8.0

sqrt- it returns the square root of a given value.

sqrt()- double

double a= Math.sqrt(4)

2.0

Q. Name any two Java Application Programming Interface packages.

Ans: i) Stand Alone

ii) Applets

Q. Mention two different styles of expressing a comment in a program.

Ans: Comments in a program can be written in the following two styles-

// <comments> -Single line Comments - Used to write comments in one line.

/* <comments>

<comments> */ Multiple line comment – Used to write comments in more than one line.

/ <comments>**

<comments> */ Documentation comment – Used to write comments in more than one paragraph.

Q. Give answer of the following:

a)	**One example of Boolean data type.**	**true**
b)	**Data type to store one letter**	**char**
c)	**Symbol used to start a block**	**{**
d)	**Command to end a case block in a switch.**	**break**
e)	**Symbol used to terminate a line in java.**	**;**

CONSTRUCTOR

Constructor: A constructor is a member function with the name same as that of the class name used to initialize the instance variables of the object.

Type of Constructor:

1. **The default constructor / non-parameterized constructor**
2. **Parameterized constructor**

Default or non-parameterized constructor:

When a class name is used as a function followed by empty brackets is known as a default or non-parameterized constructor.

Parameterized constructor:

When the class name is used as a function followed by one or list of primitive data types as parameters / arguments within the brackets is known as parameterized constructor.

The object of the class passes actual values to the arguments of the parameterized constructor.

Important guidelines while defining a constructor:

1. **A constructor function is always created by the same name as of the class i.e. class name and constructor name must be same.**
2. **The constructors are always declared and defined using public specifier. Of course the use of public is optional, private is not used before constructor creation.**
3. **A constructor has no return data type (not even void) i.e. the constructor never returns value.**
4. **A constructor without parameters is known as default/non-parameterized constructor where as a constructor with parameters is known as parameterized constructor.**
5. **The constructor cannot be called or invoked, as it automatically executes / invokes as soon as the object of the class is created and initializes values of constructor to the object itself.**
6. **Constructor is only created to initialize the data members / instance variables**

Q. Define Autoboxing and Unboxing?

Ans: The automatic conversion of primitive data type into an object of its equivalent wrapper class is known as Autoboxing.

Eg. Integer val = new Integer(26);

The integer type data 26 is converted into an object val of Integer wrapper class.

Unboxing: Unboxing is the opposite of autoboxing. It is a system of converting an object of wrapper class into primitive data type.

Eg. Integer val = new Integer(78)

int y = val;

***** The attributes of a class are represented in data members / instance variables and the behaviour is represented by member functions.**

***** A return statements can return maximum one value.**

***** OAK was the hypothetical name of Java.**

******* "An object is nothing but a variable and a variable is nothing but an object."**

******If block contains only one statement then the braces are OPTIONAL.**

******If block contains more than one statement then braces are COMPULSORY.**

DIFFERENTIATE BETWEEN

a) Differentiate between = and == in java.

	=		==
1.	It is an assignment operator which stores a value in a variable.	1.	It is a relational operator, which checks the equality between two values.

b) Differentiate between / and %

	/		%
1.	Performs division and returns the quotient.	1.	Performs division and returns the remainder.

c) Differentiate between if and switch.

	if / if else		switch
1.	It can check for a range of values.	1.	It can only check for equality with a constant value.
2.	It can work on any data type.	2.	It can only work on int and char data type.

d) Differentiate between if and if-else.

	If		if-else
1.	It is used to check conditions one by one starting from first if() to the last if(). It checks all the conditions whether any of the condition is true or false. When the control comes out of the if() then the statement given out of if() executes.	1.	It is used to check true or false condition. The statements that are associated with if( ) executes if the given condition is true otherwise the statement that are associated with else( ) executes.

Differentiate between for and while loop.

	For loop		While loop
1.	For loop contains three elements: initialization, testing, and incrementing.	1.	While loop contains only test conditions.
2.	For loop knows well in advance as to how many times the loop is to be	2.	While loop does not know that how many times the loop is to be

	executed.		**executed**
3.	**for(initialization; condition; increment)** **{** **Body** **}**	**3.**	**while(condition)** **{** **body** **}**

f) Differentiate between while and do-while / Entry control and Exit Control loops.

while / Entry control		**do-while / Exit Control loops**	
1.	**It is an entry-controlled loop, i.e. the condition is checked at the point of entry.**	**1.**	**It is an exit-controlled loop, i.e. the condition is checked at the point of exit.**
2.	**The job may not be performed even once, if the condition is false in the very first instance.** **Syntax-** **while(condition)** **{** **---------** **---------** **}**	**2.**	**The job will be performed at least once, even if the condition is false in the very first instance.** **Syntax-** **do** **{** **--------** **--------** **}** **while(condition);**

g) Differentiate between int and double.

Int		**double**	
1.	**It can store integer number**	**1.**	**It can hold decimal numbers**
2.	**It has 4 byte data size.**	**2.**	**It has 8 byte data size.**

h) Differentiate between System.out.print and System.out.println.

System.out.print()		**System.out.println()**	
1.	This statement will display the content (prompt or variable or both) and the cursor remains in the same line.	1.	This statement will display the content (prompt or variable or both) and the cursor moves to the next line.

i) State the difference between class and object.

Class		Object	
1.	A class is a collection of objects with same attributes and operations.	1.	An object is an identifiable entity which contains characteristics and behaviour.
2.	A software blue print / prototype for object are called class.	2.	An object is a software bundle of variables are related methods.
3.	An object of a particular class contains state with common behaviour.	3.	The behaviour deals with individual states of an object.
4.	A class can be called as an object factory.	4.	An object is an instance of a class.
5.	A class is a representation of only an abstraction.	5.	It is a real and unique entity have some characteristics and behaviors.

j) Distinguish between unary and binary operator.

Unary Operator		Binary Operator	
1.	Operators that act one operand are referred to as unary operator.	1.	Operators which require two operands for functioning are called binary operator.
2.	++, -- (pre-increment, pre-decrement, post-increment and post decrement), !(not) operator etc. are the example of unary operator.	2.	Mathematical operator, relational operators are the examples of binary operators.

k) Differentiate between break and continue.

Break		continue	
1.	The break statement skips the subsequent statements in a loop / switch and transfers the control out of the loop / switch block.	1.	The continue statement skips the subsequent statements in a loop and proceeds to the next iteration of the loop.
2.	Break is used in loops, switch or condition.	2.	Continue is used only in loop.

l) Differentiate between primitive and reference data types.

Primitive Data Type		Reference Data Type	
1.	Pre-defined in the Java Compiler.	1.	Defined by the programmers with reference to other data types.
2.	The fundamental or basic data type, which are independent.	2.	The derived data type which are constructed from primitive data type.
3.	Primitive data store basic data in program	3.	Reference data store memory address of an object or array.
4.	Eg. int, char, float, byte etc	4.	Eg. Array, object, String and interface.

m) Differentiate between implicit and explicit type conversion.

implicit conversion		explicit conversion	
1.	An implicit type conversion is the automatic conversion of one primitive data type into another.	1.	An explicit type conversion is forced conversion of one primitive data type into another.
2.	It does not require programmer's intervention.	2.	It requires programmer's intervention.

n) Differentiate between pre-fix and post-fix operators. OR What is the difference of ++a and a++

Prefix		Postfix	
1.	The value of the variable first changes and the changed value is used to evaluate the expression. eg-++a (change and use)	1.	The existing value of the variable is used to evaluate the expression and then the value of the variable changes. eg-a++ (use and change)

o) Differentiate between an operator and an expression in Java.

operator		expression	
1.	Operators are special symbols which perform a pre-defined task on a set of operands. eg. ++, --, *, /	1.	An expression is a combination of operators and operands which returns back a value. eg. a+b

p) Differentiate between float and double.

q) Difference	float	double
1.	It can store small floating point number	1. It can hold large floating point numbers
2.	It has 4 bytes data size.	2. It has 8 bytes data size.

ence between conditional operator and conditional statement.

Conditional Operator		Conditional Statement	
1.	A conditional operator returns back a value depending on whether a specified condition is true or false.	1.	A conditional statement performs a specific job depending on whether a specified condition is true or false.

r) Differentiate between int and double.

s) Difference	int	float
1.	It can store integer number	1. It can store decimal numbers
2.	Default value of int data type is 0	2. Default value of float data type is 0.0f

entiate between pure and impure functions

Pure Functions		Impure Functions	
1.	Pure Functions are functions in which the values of the parameter do not change.	1.	Impure function are functions in which values of parameter change.
2.	Pure function do not change the state of the object.	2.	Impure function changes the state of its object.

t) Differentiate between call by and call by reference.

Call by value		Call by reference	
1.	The formal parameter contains the copies of the actual parameter.	1.	The formal parameter contain the reference or the address of actual parameter.
2.	Changes made to the formal parameters are not reflected back in the actual parameters.	2.	Changes made to the formal parameters are automatically reflected back in the corresponding actual parameter
3.	All primitive datatypes are passed using this technique.	3.	All reference datatypes are passed using this technique.

u) Differentiate between actual parameter and formal parameter?

	Actual parameter		Formal parameter
1.	Actual parameter are parameters in the function call statement.	1.	Formal parameters are parameters in the prototype statement.

v) Difference between Void type function and Return type function.

	Void type function		Return type function
1.	A void type function does not return back any value to the calling function.	1.	A return type function returns back a value of the specified data type to the calling function.

w) Difference between Bubble Sort and Selection Sort.

	Bubble Sort		Selection Sort
1.	It takes more time as compared to selection sort as it takes more iteration to complete the sorting.	1.	It takes less time as compared to bubble sort as it takes less iteration to complete the sorting.
2.	The bubble sorting process checks adjacent elements of the array, i.e. first element with the second element, the second element with the third element and the process continues until all elements are exhausted. In case the array element is greater or lesser than the adjacent element (for ascending or descending order), the elements are interchanged immediately. The process continues until all elements are sorted.	2.	During the selection sort process, an element is taken from the first and from the remaining elements the smallest or largest number (for ascending or descending order respectively) is found and interchanged.

x) Difference between linear search and binary search.

	Linear Search		Binary Search
1.	Linear Search works with both sorted and unsorted array.	1.	Binary search works only on sorted array.
2.	Linear Search is a search process technique that involves checking each element sequentially with the given element to be searched for.	2.	Binary search is a search process technique where an element to be searched is checked with the central element of the array and depending upon its value, it is checked in the sub-array before the central position or in the sub-array after the central position. The same search process continues even with the sub-array.

y) Differentiate between Constructor and Method.

Constructor		Function / Method	
1.	It is called automatically when an object is created.	1.	It is called explicitly when need to work.
2.	It has same name as its class name.	2.	Any name as per programmer choice other than class name.
3.	Constructor is specified as public.	3.	Method may be public, private or protected.
4.	It does not return any value it only initialize the value into instance variable.	4.	It may return or may not return any value.

z) Differentiate between Local Variable and Instance Variable.

Local Variable		Instance Variable	
1.	Variable which is declare in a scope or function is called local variable.	1.	Instance variables of a class come into existence when object of class is created. There is one copy of instance member for each and every object.
2.	The variable has specific scope to access.	2.	Generally declare inside class not in a method.
3.	Access visibility mode is limited	3.	Access visibility mode inside class.

aa) Differentiate between length and length().

length		length()	
1.	length is used to find the number of elements in an array	1.	length() is used to find the number of characters in a string.

bb) Differentiate between indexOf() and charAt()

indexOf()		charAt()	
1.	It returns the index number of the first occurrence of a character in a string.	1.	It returns the character present at a specified index number.
2.	Its return type is int.	2.	Its return type is char.

cc) Differentiate between Autoboxing and Unboxing.

	Autoboxing		Unboxing
1.	Automatic conversion of primitive types (int) to their corresponding object wrapper class (Integer) is known as boxing.	1.	Converting an object of a wrapper lass (Integer) to its corresponding primitive type (int) is known as unboxing.
2.	It applies when a primitive value is passed as a parameter to a method that expect an object of the corresponding wrapper class.	2.	It applies when an object of a wrapper class is passed as a parameter to a method that expects a value of the corresponding primitive type.

JAVA THEORY

Q. What is a class?

Ans: A class is a blueprint or prototype of a group of objects having similar characteristic and behaviour.

Q. What is object?

Ans: An object is an instance of a class. It possesses all the characteristics and behaviour as defined in a class.

Q. What are the characteristics / features of Java?

Ans: Java is a case sensitive language.

It is Platform independent language.

It is an object oriented language.

It is a real-life oriented language.

It uses re-usability.

It uses readability

Java is also a compiler language.

Java is a structured language.

Q. Define Object Oriented Programming (OOP).

Ans: An Object Oriented Programming (OOP) is a modular approach which allows the data to be applied within stipulated program area. It is also provided the reusability feature to develop productive logic which means to give more emphasis on data. It has been developed to increase the programmer's productivity and also to overcome the weaknesses of traditional approach of programming.

Q. Name the basic elements of OOP (Principles of OOP). / Name some OOP principles.

Ans: The basic elements of the Object oriented programming (OOP) are: Class, Object, Data Abstraction, Encapsulation, Inheritance, Polymorphism, Data Hiding.

Q. Concept of objects and class.

Ans: A class and an object have a close relationship in Object Oriented Programming. Class is used to create various objects which possess same characteristics and behaviour within the class. So we can say that class is a blue print or prototype of an object. The characteristics and behaviour of the class is declared at the time of its creation. Each object of the class possesses same features defined within the class. e.g. If Car is the class with the characteristics / make (name of the company), colour, model and version, then a car, Maruti which is white in colour, Swift DZire, diesel version is used as an object of the class 'Car'.

Q. What is Abstraction?

Ans: Abstraction is the act of representing data members of a class without knowing the background details.

Q. What do you mean by Encapsulation?

Ans: Encapsulation is a feature of JAVA by which we wrap-up data and functions, into one single unit known as CLASS.

Q. Define Polymorphism.

Ans: Polymorphism is a feature of Java, by which one object can behave differently under different circumstances. It is implemented in java using function overloading.

Q. Define Inheritance.

Ans: Inheritance is a feature of Java by which one class can acquire the properties of another class. The class whose properties are acquired are known as base class / super class and the class which acquires the properties of the base class is known as derived or sub-class.

Q. What is a class and an object with reference to java?

Ans: A class is a blueprint or prototype of a group of objects having similar characteristic and behaviour.

An object is an instance of a class. It possesses all the characteristics and behaviour as defined in a class.

Q. Why class is called an object factory?

Ans: A class is called an object factory as it is a template, using which any no of objects can be created.

Q. Why is a class known as composite datatype?

Ans: A composite data type is one which is composed with various primitive data type. A class defined with various primitive data types such as int, double etc, so it is known as composite data type.

Q. Why is an object also known as an instance of a class?

Ans: Since, an object possesses instant variables and member method defined within the class. This is the reason why an object is called an instance of a class.

Q. Define Byte code.

Ans: It is a set of pseudo language instructions which are directly interpreted by the JVM thereby making java a platform independent language.

Q. Define the term Platform independence.

Ans: A java program is compiled into a byte code. A byte code is a pseudo-machine language instruction, which is directly interpreted by the JVM (Java Virtual Machine), thereby making Java a platform independent language. Platform independence program can run identically on Windows and Linux.

Q. Explain "Java is a Case Sensitive language".

Ans: Java is a case sensitive language as the upper case and lower case letters are distinguished by the language. It treats upper case variable and lower case variable differently.

Q. What do you mean by Structure of a Java program?

Ans: A java program is a block / blocks inside a block.

Q. Define Block.

Ans: A Block is a group of statement or function enclosed between { }. A compound statement is also called a block.

Q. Define JVM.

Ans: JVM - Java virtual Machine

Q. Define SCOPE.

Ans: The SCOPE of a variable is the range of validity of a variable. A variable is valid only in the Block, in which it is declared.

Q. Define Tokens.

Ans: A token is the smallest individual unit of a program that is meaningful to the java compiler.

Q. What are the different types of tokens?

- **Identifiers**
- **Literals**
- **Separators -, &, ;**
- **Operators +, -, ++, >=, <=**
- **Keywords**

Q. Define Identifiers.

Ans: Identifiers: are names which are defined by the programmer in the program.

In java we have 3 identifiers -

1. **Class name**
2. **Function / Method name**
3. **Variable name**

Q. What are the rules of identifiers in Java?

Ans: It should start with an alphabet ONLY.

It can be any length (upto 256 characters)

Dollar and underscore are the only two symbols allowed in Java (anywhere after the first character)

Q. Define Literals. Explain each literals with example.

Ans: The constant used in a java programs which remain fixed throughout the program is called literals or constants.

Integer Literals: The numbers which are represented without decimal point are called Integer Literals. These are the whole numbers having positive or negative values. e.g. 14, 5

Real Literals: Real Literals are also called floating point constants. They represent numbers with decimal point. e.g. 24.6, 0.007.

Character Literals: The constants, which are alphanumeric in nature, are called character literals. All alphabets upper or lower case, digits special characters can be terms as character literals. e.g. 'A', 'd', '3', '*' etc

String Literals: String is a set of alphanumeric characters. A group of characters enclosed within a pair of opening and closing double quotes is known as string literals. e.g. "Ajay", "games"

Q. What do you mean by an operator? Give two examples of mathematical operator.

Ans: Operators are special symbols which perform a pre-defined task on a set of operands. + and * are two mathematical operators.

TYPES OF OPERATORS:

1) **ARITHMETICAL OPERATORS**
2) **RELATIONAL OPERATORS**
3) **LOGICAL OPERATORS**

Arithmetical Operators: The operators, which are applied to perform arithmetical calculations in a program are known as arithmetical operators.

Types of arithmetical operator,

- **Unary operators (++, --): An arithmetical operator, which is applied with a single operand. Unary Increment Operators (++) increases the value of an operand by one whereas Unary Decrement Operators (--) decreases the value of an operand by one.**

  PREFIX: The value of the variable first changes and then the changed value is used to evaluate the expression. Eg: ++a (change and use)

  POSTFIX: The existing value of the variable is used to evaluate the expression & then the value of the variable changes. Eg: b++ (Use and change)

- **Binary operators (+, -, *, /, %): An arithmetic operator, which deals with two operands is known as Binary Arithmetic Operator.**

ARITHMETICAL OPERATORS	SYMBOLS	
1. Multiply...	*	Same Preference – Left to Right
2. Divide	/	
3. REMAINDER OR MODULUS	%	
4. Add	+	Same Preference – Left to Right
5. Subtract	-	

RELATIONAL OPERATORS: These operators are used to show the relationship among the operands. Relational operators compare the values of the variables and result in 'True' or 'False".

The different types of Relational Operators are as follows:

RELATIONAL OPERATORS	SYMBOLS
1. Greater than	>
2. Greater than equal to	>=
3. Less than	<
4. Less than equal to	<=
5. Equal to	==
6. Not equal to	!=

LOGICAL OPERATORS: Java uses logical operators AND (&&), OR (||) or NOT (!). Logical operators are used when we want to check multiple conditions together. We can combine many relational express using Logical and operations. The result will be a boolean type.

LOGICAL OPERATORS	SYMBOLS
1. NOT	!
2. AND	&&
3. OR	||

Logical AND (&&): The AND operator results in true if both the expression (comprising its operands) are true.

5>3 && 3<5, Results in true because both the expressions are true.

6==6 && 3>0, Results in true because both the expressions are true.

5! =5 && 4==4, Results in false as the first expression is false.

Logical OR (||): The OR operator is used to combine two conditional expressions. It will result in true if either of the two conditions (expression) is true otherwise false.

5>4 || 8>12, Results in true because 5>4 true.

3>7 || 5<=4, Results in false because both the expressions are false

2<0 || 2<12, It will result in true because second expression is true.

Logical NOT (!): Logical NOT operator is applied when you want to revert the outcome of an expression. It is a unary operator because it uses a single operand.

!(8>3), False, because 8>3 is true

!(5<7), False, as 5<7 is true

!(3<0), True as 3<0 is false

JAVA SHORTHAND ASSIGNMENT OPERATORS

x+=10	**equivalent to**	**x=x+10**
x-=10	**equivalent to**	**x=x-10**
x*=3	**equivalent to**	**x=x*3**
x/=2	**equivalent to**	**x=x/2**
x%=z	**equivalent to**	**x=x%z**

Q. Define Precedence of operators.

Ans: The order in which the operators are evaluated in an expression is known as precedence of operators. e.g. ++, %, >=, &&

Q. What is a ternary operator? Give an example.

Ans: The conditional operator is also known as the ternary operator because it operates on three operands.

Syntax: condition ? true value : false value

Example 1: int x = (y==24) ? 400 : 200;

Output : Here if the value of variable y is 24, then variable x stores 400 otherwise 200.

Q. **Why is the conditional operator also known as the ternary operator?**

Ans: **The conditional operator is also known as the ternary operator because it operates on three operands.**

Syntax: **condition ? true value : false value**

Q. **Define Keywords. Name any two keywords in Java.**

Ans: **Keywords are words which are predefined for the java compiler and have a special meaning for the java compiler. public, static, void, import**

Q. **What is the role of the new keyword?**

Ans: **The keyword new is used to allocate space in the dynamic memory for the storage of data and functions belonging to an object in java programming. It creates a new instance of a class.**

Q. **What is the role of dot operator?**

Ans: **The dot operator facilitates invoking members of the class to carry out the tasks. Example : java.util.*;**

Q. **What is a variable? Give two examples.**

Ans: **A Variable is a named memory location which stores a value of a specified data type. Example: int a, double b.**

Q. **What is an expression? Give an example.**

Ans: **An expression is a combination of operators and operands which returns back a value. Ex: a+b**

Q. **Define Statement.**

Ans: **If an expression is assigned to a variable, it is known as statement.**

c=a+b;

Q. **What is a compound statement?**

Ans: **Compound statements are statements that contain lists of statements enclosed in braces { }**

Example:

```
{
        statement;
        statement;
}
```

Q. What are the two types of data in java?

Ans: There are mainly two types of data:

Primitive data are pre-defined or inbuilt in the java compiler

Reference data are defined by the programmer with reference to other data types.

Types of Data

i) Primitive		**ii) Reference**	
1	**char – 1 character 'A', '8', '!'**	**1**	**classes**
2	**byte – integers**	**2**	**objects**
3	**short – integers**	**3**	**arrays**
4	**int – whole number**	**4**	**interfaces**
5	**long – ten digit numbers**		
6	**float – real numbers v, decimal numbers**		
7	**double – real numbers, decimal numbers**		
8	**boolean – true / false**		

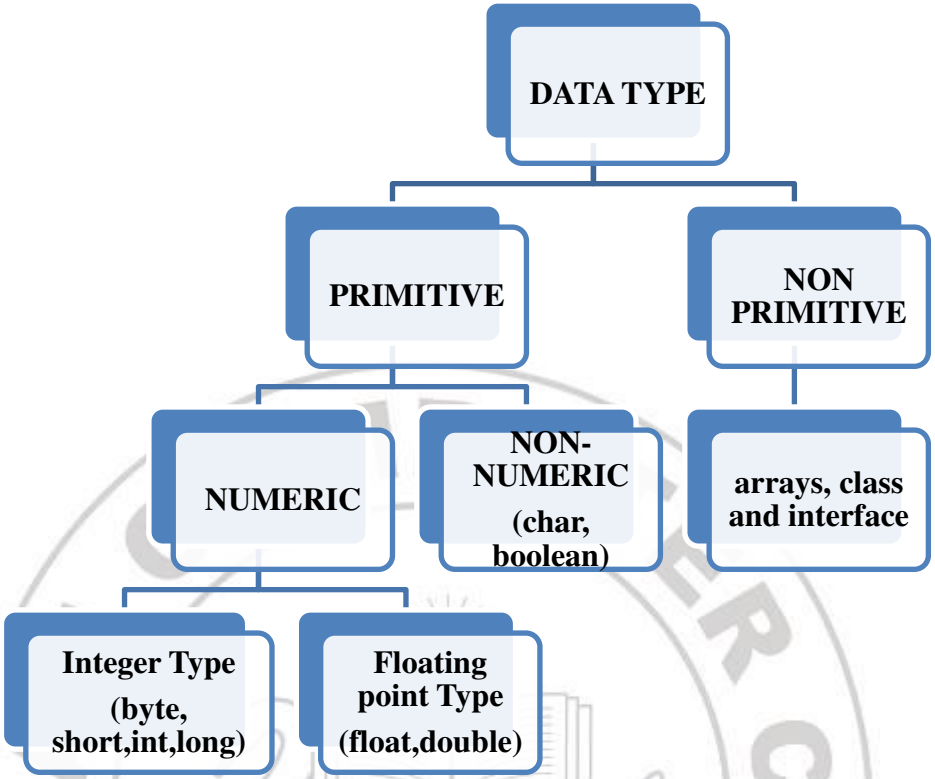
Q. Define Data Type.

Ans: The data type represents that the particular entity is used for which kind of operation.

The data types are divided into two categories:

1) **Primitive or pre-defined data types: These data types are available with Java language. They are transferred in the memory as soon as Java program is loaded in the computer system. The primitive data types supported by Java are: byte, short, int, long, float, double, char, boolean.**

2) **Non-Primitive or Derived or Reference data types: These data types are formed with the help of primitive data types. The non-primitive data types supported by java are: arrays, class and interface.**
A non-primitive data type or reference data type is used to store the memory address of an object.



Q. What are the default values of int and float?

Ans:

Data type	Size	Range of values	Default Value
char	2 bytes		'\u0000'
byte	1 byte	-2^7 to 2^7-1(-128 to 127)	0
short	2 bytes	-2^{15} to $2^{15}-1$(-32768 to 32767)	0
int	4 bytes	-2^{31} to $2^{31}-1$	0
long	8 bytes	-2^{63} to $2^{63}-1$	0 l
float	4 bytes	-3.4×10^{-38} to 3.4×10^{38}	0.0f
double	8 bytes	$-1.7 \times e^{-308}$ to $1.7 \times e^{308}$	0.0d
boolean	1 bit		false

8 bits=1byte
$e^2 = 10^2$

$e^{-2} = 10^{-2}$

Q. What are the values of a Boolean data type?

Ans: Boolean data type either stores true or false values.

Q. Name 2 decision making statements in java.

Ans: Two decision making statements in java are

a) if – else if b) switch case

Q. What is the function of the import statement?

Ans: The import statement helps us to use or include classes of an external package.

import package.class

Q. What is the role of default in a switch statement?

Ans: The default statement is executed if the switch statement does not match any of the given case. In the absence of the default statement in a switch no job is performed if the switch variable does not match any of the given cases.

Q. What happens if a default is not included in a switch statement?

Ans: If default is not included in a switch statement and if the user is inputting any such value which does not match with any of the given case then no job is performed.

Q. What is the purpose of break in a switch statement?

Ans: The break statement skips the subsequent statements in a loop / switch and transfers the control out of the loop / switch block.

Q. What do you understand by fall through with respect to switch case?

Ans: The switch statement has a fall through property by which it keeps executing all the subsequent cases till it completes executing the entire switch block. To prevent this a break statement is given after the case thereby transferring the control out of the switch block on the completion of desired case.

The default statement is executed if the switch statement does not match any of the given case. In the absence of the default statement in a switch no job is performed if the switch variable does not match any of the given cases.

Q. Define multiple branching statement in java.

Ans: The multiple branching statement in Java is switch. Switch statement works on the basis that the value of a variable is.

Q. Is it compulsory to use a default in a switch? What will happen if default statement is not given?

Ans: It is not compulsory to use a default in a switch. In the absence of a default statement, no job is performed if the switch variable does not match any of the given cases.

Q. Name any 3 jump statement in java. Explain each of them.

Ans: Java provides three jumps statements:

1. **Return: It is a keyword and this statement returns back the program control from a called function to its calling functions.**
2. **Break: Break statement terminates the control from the current loop. This statement is appear inside the loop and switch case statement. This statement skips the rest of the loop and jumps over the statement following the loop and switch case.**
3. **Continue: This statement abandon the current iteration of the loop by skipping over the rest of the statements in the loop body and immediately transfer the control to the evolution of the test expression of the loop for the next iteration of the loop.**

Exception Handling

Q. Define Exception.

Ans: Exception: An exception is an anomalous situation / error which may occur during the execution of the program / runtime.

Q. What is an exception handler?

Ans: Exception Handler: An exception handler is a class which contains the definition of all possible anomalous situation that may arise during runtime, and the cause of action to be taken there upon. The job of the exception handler is not only to ensure the smooth functioning of the program but also the smooth execution of it. eg. Example - throws, try, catch.

Q. Name 2 Exception handlers.

Ans: Try and catch are two exception handlers: The normal processing sequence to be performed is included in the try block. The sequence of instruction to be performed on the occurrence of an exception included in the catch block

Q. What do you mean by type casting?

Ans: Type casting is the forced conversion of one primitive data type into another using the Type caste operator (). The type caste operator is () with the data type to be converted into, being specified within ().

Q. What is type conversion? Explain the two types of type conversion.

Ans: The process of converting one primitive data type into another is called Type Conversion. Java facilitates the type conversion in two forms:

i. Implicit type conversion: An implicit type conversion is a conversion performed by the compiler automatic without programmer's intervention. An implicit conversion is applied generally whenever differing data types are intermixed in an expression (mixed mode expression), so as not to lose information.
e.g. int a; double b; double c;
c=a+b;

ii. Explicit type conversion: An explicit type conversion is user-defined that forced an expression to be of specific type with programmer's intervention.
e.g.: int a; int b; double c;
b= (int) (a*c);

Loop

Q. What is loop? Name any one.

Ans: Loop – A loop is an iteration statement, which repeats a job a required number of times.

```
for (initial expression; test expression; update expression).

for (int i=1; i<=5; i++)

{

        System.out.println (i);

}
```

Q. What is the infinite loop? Give an example.

Ans: An infinite loop is a loop which never terminates.

EX-1

```
for(;;)
{
        System.out.println ("LMB");
}
```

EX-2

```
while (true)
{
        System.out.println ("LMB");
}
```

Q. Define Dummy Loop.

Ans: If we use semicolon (;) after for loop it means for loop does not have its body and it has no statement inside the loop. The loop will initialize and terminate within its bracket. This type of loop is called DUMMY LOOP. eg.- for (int i=1; i<=10; i++);

Q. Define Conditional loops

Ans: Conditional loops repeat a given job as long as a specified condition is true.

Q. What is the role of do in a do-while loop?

Ans: The 'do' statement marks the beginning of the do while loop. It executes the block of codes at least once in the do-while loop without checking any condition.

Q. What is the use of a for loop? Explain the for loop with an example.

Ans: The for loop is used for fixed iterations in the program.

Syntax- for (initial expression; test expression; update expression)

```
{
    statements;
}
```

Example

```
class test
{
    public void(int n)
    {
            int a, sum;
            for (a=1; a<=n; a++)
            {
                System.out.println (a);
                sum = sum+a;
            }
    }
}
```

Q. Explain the while loop with an example.

Ans: while (): This loop is used to process one or more than one statements till the conditions is true/satisfies and the loop terminates as soon as the condition becomes false.

The while loop is also known as Entry Controlled Loop because before entering into the loop first the condition is checked and if it is true, the control enters into the loop and executed all the statements given within curly braces of while, if the entry conditions is false, the loop terminates and executes those instructions given out of the while loop.

Syntax:

```
while (test/ending condition)
{
statement;
statement;
-------
Increment/decrement statement or update statement
}
```

Q. Write the syntax and one example of if-else.

Ans: Eg. if (a= =10)

```
        System.out.println ("Executes")
else
        System.out.println ("Not executes")
```

A Conditional Statement in java is 'if'

Syntax

```
if (condition)
true; ob
else
false; ob
```

Q. Define Static and Non-static Method.

Ans: Static – In order to access a class member declared as static, an instance of the class need not be created, as the member is directly accessible to the class.

Non-static:- In order to access a non-static class member, an instance of the class has to be created, as the member is accessible only through the instance.

Q. Write a Java statement to create an object lamartiniere of the class School.

Ans: School lamartiniere = new School ( );

Q. Write the assignment statement for x+=2.

Ans: x=x+2;

LIBRARY CLASS

Q. Define Library class.

Ans: Library class – Library classes are classes which are predefined or inbuilt in JAVA.

Data InputStream

BufferedReader & InputStreamReader

Scanner

Q. Define package

Ans: Package – A package is a collection of related classes and interfaces.

Creating Objects.

Syntax:- classname objectname = new classname

The word new creates a new instance of a class

Relation of BufferedReader & InputStreamReader:_

The object of InputStreamReader is taken as the parameter of BufferedReader.

InputStreamReader read = new InputStreamReader (System.in)

BufferedReader br = new BufferedReader (read)

or

BufferedReader br=new BufferedReader (new InputStreamReader (System.in);

Q. Name the Java keyword that:

a) Converts a String to primitive int datatype.
b) Creates a new instance of a class.

Ans: a) Integer.parseInt(s)

b) classname objectname=new classname();

Q. Name the keywords that:

a) returns the control to the calling function.
b) Allocates memory space for an object.

Ans: a) return

b) new

Q. Define Temporary object.

Ans: A temporary object is an object which gets created to perform its job and then ceases to exist.

Q. What is the function of readLine?

Ans: ReadLine: - It accepts the value from the user in string form.

Integer.parseInt ():- It converts a String to int.

For

int :- Integer.parseInt();

double :- Double.parseDouble();

float :- Float.parseFloat();

String :- br.readLine();

Wrapper Classes

Q. Define Wrapper class..

Ans: Wrapper classes are classes which wrap up a primitive data type.

OR

Wrapper classes are classes which help us to create objects of a primitive data type.

Data Type	Wrapper Class
byte	Byte
char	Character
short	Short
int	Integer
float	Float
double	Double
long	Long
boolean	Boolean

char x= (char)br.read(); statement

char x= sc.nextLine().charAt(0); function

Q. **Write the statement to input a character in java.**

Ans: **char x=sc.next().charAt(0);**

Q. **Write a statement to convert string into int.**

Ans: **int a=Integer.parstInt(str)**

Q. **Write an input statement for float.**

Ans: **float b=Float.parseFloat(str).**

Q. **How many types of errors are there?**

Ans: **There are three types of errors.**

Syntax / compiled time error

Runtime error

Logical error

Q. **Define Access Specifiers.**

Ans: **Access specifiers are keywords which specify the accessibility of a class. i.e places from where a member class is accessible.**

Q. **Define all access Specifiers.**

Ans: **PUBLIC : Public is an access specifier which can be accessed by the classes and sub classes of the same package as well as the classes and the sub classes of the other packages**

PRIVATE: Private is an access specifier which can be accessed only by the classes in which it is created and neither by the sub classes of the same package nor by the classes and the sub classes of the other packages

PROTECTED: Protected is an access specifier which can be accessed by the classes and sub classes of the same package but not by the classes and the sub classes of the other packages

	SAME PACKAGE		OTHER PACKAGE	
	CLASSES	SUB-CLASSES	CLASSES	SUB-CLASSES
PUBLIC	YES	YES	YES	YES
PRIVATE	YES	X	X	X
PROTECTED	YES	YES	X	X

FUNCTION

Q. What are the arguments of a method? How many arguments can a method contain?

Ans: The argument of a function specify the number and type of values to be passed to a function in order to perform its job. The argument of a method is a comma-separated list of variables of a method referred to as its arguments or parameters. A methods / functions may be without any parameters, in which case the parameter list is empty.

Q. Define function prototype statement?

Ans: The function prototype statement is the first line of the function, which denotes the return type of the function, function name and the number and types of parameters to the function.

void	**area**	**int a, int b**
return type	**name**	**parameters / arguments**

Q. Why do we use void in function prototype statement?

Ans: The void keyword is given to a function prototype statement to meet the demand of the syntax. The void keyword signifies that the function does not return back any value to the calling function.

Q. What is the role of the void keyword in a function prototype statement?

Ans: The void keyword signifies that the function does not return back any value to the calling function.

Q. How many values can a function return?

Ans: A void type function does not return any value whereas non-void type / with return type function returns only 1 value.

Q. What are the uses / advantages of using functions?

Ans: The followings are the uses / advantages of functions:-

a) **Functions reduces complexity of code.**
b) **Functions gives program a better structure.**
c) **Functions enhances re-usability of code.**
d) **Functions hides implementation details.**

Q. Write the function prototype for the function "sum" that takes an integer variable (x) as its argument and returns a value of float datatype.

Ans: float sum(int x)

Q. What do you mean by invoking a function? What are the ways of invoking functions in java?

Ans: A process of using a method in a program is referred as calling a method or invoking a method. Call by value and call by reference are the two ways of invoking functions.

Q. Define Function Overloading.

Ans: Function overloading is a situation in which a class has more than one function having the same name but different parameter list / argument list / signature.

Q. Which OOP principle implements function overloading?[2007]

Ans: Polymorphism implements function overloading.

Q. Define instance variable.

Ans: An instance variable which is present in an instance of the class. i.e. an object.

Function of pow()

The pow function returns the value of a no. raised to the power of another no.

eg. double c=Math.pow(2,3)

8.0

sqrt- it returns the square root of a given value.

sqrt()- double

double a= Math.sqrt(4)

2.0

Q. Name any two Java Application Programming Interface packages.

Ans: i) Stand Alone

ii) Applets

Q. Mention two different styles of expressing a comment in a program.

Ans: Comments in a program can be written in the following two styles-

// <comments> -Single line Comments - Used to write comments in one line.

/* <comments>

<comments> */ Multiple line comment – Used to write comments in more than one line.

/ <comments>**

<comments> */ Documentation comment – Used to write comments in more than one paragraph.

Q. Give answer of the following:

a)	**One example of Boolean data type.**	**true**
b)	**Data type to store one letter**	**char**
c)	**Symbol used to start a block**	**{**
d)	**Command to end a case block in a switch.**	**break**
e)	**Symbol used to terminate a line in java.**	**;**

CONSTRUCTOR

Constructor: A constructor is a member function with the name same as that of the class name used to initialize the instance variables of the object.

Type of Constructor:

1. **The default constructor / non-parameterized constructor**
2. **Parameterized constructor**

Default or non-parameterized constructor:

When a class name is used as a function followed by empty brackets is known as a default or non-parameterized constructor.

Parameterized constructor:

When the class name is used as a function followed by one or list of primitive data types as parameters / arguments within the brackets is known as parameterized constructor.

The object of the class passes actual values to the arguments of the parameterized constructor.

Important guidelines while defining a constructor:

1. **A constructor function is always created by the same name as of the class i.e. class name and constructor name must be same.**
2. **The constructors are always declared and defined using public specifier. Of course the use of public is optional, private is not used before constructor creation.**
3. **A constructor has no return data type (not even void) i.e. the constructor never returns value.**
4. **A constructor without parameters is known as default/non-parameterized constructor where as a constructor with parameters is known as parameterized constructor.**
5. **The constructor cannot be called or invoked, as it automatically executes / invokes as soon as the object of the class is created and initializes values of constructor to the object itself.**
6. **Constructor is only created to initialize the data members / instance variables**

Q. Define Autoboxing and Unboxing?

Ans: The automatic conversion of primitive data type into an object of its equivalent wrapper class is known as Autoboxing.

Eg. Integer val = new Integer(26);

The integer type data 26 is converted into an object val of Integer wrapper class.

Unboxing: Unboxing is the opposite of autoboxing. It is a system of converting an object of wrapper class into primitive data type.

Eg. Integer val = new Integer(78)

int y = val;

***** The attributes of a class are represented in data members / instance variables and the behaviour is represented by member functions.**

***** A return statements can return maximum one value.**

***** OAK was the hypothetical name of Java.**

******* "An object is nothing but a variable and a variable is nothing but an object."**

******If block contains only one statement then the braces are OPTIONAL.**

******If block contains more than one statement then braces are COMPULSORY.**

DIFFERENTIATE BETWEEN

a) Differentiate between = and == in java.

	=		==
1.	It is an assignment operator which stores a value in a variable.	1.	It is a relational operator, which checks the equality between two values.

b) Differentiate between / and %

	/		%
1.	Performs division and returns the quotient.	1.	Performs division and returns the remainder.

c) Differentiate between if and switch.

	if / if else		switch
1.	It can check for a range of values.	1.	It can only check for equality with a constant value.
2.	It can work on any data type.	2.	It can only work on int and char data type.

d) Differentiate between if and if-else.

	If		if-else
1.	It is used to check conditions one by one starting from first if() to the last if(). It checks all the conditions whether any of the condition is true or false. When the control comes out of the if() then the statement given out of if() executes.	1.	It is used to check true or false condition. The statements that are associated with if( ) executes if the given condition is true otherwise the statement that are associated with else( ) executes.

Differentiate between for and while loop.

	For loop		While loop
1.	For loop contains three elements: initialization, testing, and incrementing.	1.	While loop contains only test conditions.
2.	For loop knows well in advance as to how many times the loop is to be	2.	While loop does not know that how many times the loop is to be

	executed.		**executed**
3.	**for(initialization; condition; increment)** **{** **Body** **}**	**3.**	**while(condition)** **{** **body** **}**

f) Differentiate between while and do-while / Entry control and Exit Control loops.

while / Entry control		**do-while / Exit Control loops**	
1.	**It is an entry-controlled loop, i.e. the condition is checked at the point of entry.**	**1.**	**It is an exit-controlled loop, i.e. the condition is checked at the point of exit.**
2.	**The job may not be performed even once, if the condition is false in the very first instance.** **Syntax-** **while(condition)** **{** **---------** **---------** **}**	**2.**	**The job will be performed at least once, even if the condition is false in the very first instance.** **Syntax-** **do** **{** **--------** **--------** **}** **while(condition);**

g) Differentiate between int and double.

Int		**double**	
1.	**It can store integer number**	**1.**	**It can hold decimal numbers**
2.	**It has 4 byte data size.**	**2.**	**It has 8 byte data size.**

h) Differentiate between System.out.print and System.out.println.

System.out.print()		**System.out.println()**	
1.	This statement will display the content (prompt or variable or both) and the cursor remains in the same line.	1.	This statement will display the content (prompt or variable or both) and the cursor moves to the next line.

i) State the difference between class and object.

Class		Object	
1.	A class is a collection of objects with same attributes and operations.	1.	An object is an identifiable entity which contains characteristics and behaviour.
2.	A software blue print / prototype for object are called class.	2.	An object is a software bundle of variables are related methods.
3.	An object of a particular class contains state with common behaviour.	3.	The behaviour deals with individual states of an object.
4.	A class can be called as an object factory.	4.	An object is an instance of a class.
5.	A class is a representation of only an abstraction.	5.	It is a real and unique entity have some characteristics and behaviors.

j) Distinguish between unary and binary operator.

Unary Operator		Binary Operator	
1.	Operators that act one operand are referred to as unary operator.	1.	Operators which require two operands for functioning are called binary operator.
2.	++, -- (pre-increment, pre-decrement, post-increment and post decrement), !(not) operator etc. are the example of unary operator.	2.	Mathematical operator, relational operators are the examples of binary operators.

k) Differentiate between break and continue.

Break		continue	
1.	The break statement skips the subsequent statements in a loop / switch and transfers the control out of the loop / switch block.	1.	The continue statement skips the subsequent statements in a loop and proceeds to the next iteration of the loop.
2.	Break is used in loops, switch or condition.	2.	Continue is used only in loop.

l) Differentiate between primitive and reference data types.

Primitive Data Type		Reference Data Type	
1.	Pre-defined in the Java Compiler.	1.	Defined by the programmers with reference to other data types.
2.	The fundamental or basic data type, which are independent.	2.	The derived data type which are constructed from primitive data type.
3.	Primitive data store basic data in program	3.	Reference data store memory address of an object or array.
4.	Eg. int, char, float, byte etc	4.	Eg. Array, object, String and interface.

m) Differentiate between implicit and explicit type conversion.

implicit conversion		explicit conversion	
1.	An implicit type conversion is the automatic conversion of one primitive data type into another.	1.	An explicit type conversion is forced conversion of one primitive data type into another.
2.	It does not require programmer's intervention.	2.	It requires programmer's intervention.

n) Differentiate between pre-fix and post-fix operators. OR What is the difference of ++a and a++

Prefix		Postfix	
1.	The value of the variable first changes and the changed value is used to evaluate the expression. eg-++a (change and use)	1.	The existing value of the variable is used to evaluate the expression and then the value of the variable changes. eg-a++ (use and change)

o) Differentiate between an operator and an expression in Java.

operator		expression	
1.	Operators are special symbols which perform a pre-defined task on a set of operands. eg. ++, --, *, /	1.	An expression is a combination of operators and operands which returns back a value. eg. a+b

p) Differentiate between float and double.

q) Difference	float	double
1.	It can store small floating point number	1. It can hold large floating point numbers
2.	It has 4 bytes data size.	2. It has 8 bytes data size.

ence between conditional operator and conditional statement.

Conditional Operator		Conditional Statement	
1.	A conditional operator returns back a value depending on whether a specified condition is true or false.	1.	A conditional statement performs a specific job depending on whether a specified condition is true or false.

r) Differentiate between int and double.

s) Difference	int	float
1.	It can store integer number	1. It can store decimal numbers
2.	Default value of int data type is 0	2. Default value of float data type is 0.0f

entiate between pure and impure functions

Pure Functions		Impure Functions	
1.	Pure Functions are functions in which the values of the parameter do not change.	1.	Impure function are functions in which values of parameter change.
2.	Pure function do not change the state of the object.	2.	Impure function changes the state of its object.

t) Differentiate between call by and call by reference.

Call by value		Call by reference	
1.	The formal parameter contains the copies of the actual parameter.	1.	The formal parameter contain the reference or the address of actual parameter.
2.	Changes made to the formal parameters are not reflected back in the actual parameters.	2.	Changes made to the formal parameters are automatically reflected back in the corresponding actual parameter
3.	All primitive datatypes are passed using this technique.	3.	All reference datatypes are passed using this technique.

u) Differentiate between actual parameter and formal parameter?

	Actual parameter		Formal parameter
1.	Actual parameter are parameters in the function call statement.	1.	Formal parameters are parameters in the prototype statement.

v) Difference between Void type function and Return type function.

	Void type function		Return type function
1.	A void type function does not return back any value to the calling function.	1.	A return type function returns back a value of the specified data type to the calling function.

w) Difference between Bubble Sort and Selection Sort.

	Bubble Sort		Selection Sort
1.	It takes more time as compared to selection sort as it takes more iteration to complete the sorting.	1.	It takes less time as compared to bubble sort as it takes less iteration to complete the sorting.
2.	The bubble sorting process checks adjacent elements of the array, i.e. first element with the second element, the second element with the third element and the process continues until all elements are exhausted. In case the array element is greater or lesser than the adjacent element (for ascending or descending order), the elements are interchanged immediately. The process continues until all elements are sorted.	2.	During the selection sort process, an element is taken from the first and from the remaining elements the smallest or largest number (for ascending or descending order respectively) is found and interchanged.

x) Difference between linear search and binary search.

	Linear Search		Binary Search
1.	Linear Search works with both sorted and unsorted array.	1.	Binary search works only on sorted array.
2.	Linear Search is a search process technique that involves checking each element sequentially with the given element to be searched for.	2.	Binary search is a search process technique where an element to be searched is checked with the central element of the array and depending upon its value, it is checked in the sub-array before the central position or in the sub-array after the central position. The same search process continues even with the sub-array.

y) Differentiate between Constructor and Method.

Constructor		Function / Method	
1.	It is called automatically when an object is created.	1.	It is called explicitly when need to work.
2.	It has same name as its class name.	2.	Any name as per programmer choice other than class name.
3.	Constructor is specified as public.	3.	Method may be public, private or protected.
4.	It does not return any value it only initialize the value into instance variable.	4.	It may return or may not return any value.

z) Differentiate between Local Variable and Instance Variable.

Local Variable		Instance Variable	
1.	Variable which is declare in a scope or function is called local variable.	1.	Instance variables of a class come into existence when object of class is created. There is one copy of instance member for each and every object.
2.	The variable has specific scope to access.	2.	Generally declare inside class not in a method.
3.	Access visibility mode is limited	3.	Access visibility mode inside class.

aa) Differentiate between length and length().

length		length()	
1.	length is used to find the number of elements in an array	1.	length() is used to find the number of characters in a string.

bb) Differentiate between indexOf() and charAt()

indexOf()		charAt()	
1.	It returns the index number of the first occurrence of a character in a string.	1.	It returns the character present at a specified index number.
2.	Its return type is int.	2.	Its return type is char.

cc) Differentiate between Autoboxing and Unboxing.

	Autoboxing		Unboxing
1.	Automatic conversion of primitive types (int) to their corresponding object wrapper class (Integer) is known as boxing.	1.	Converting an object of a wrapper lass (Integer) to its corresponding primitive type (int) is known as unboxing.
2.	It applies when a primitive value is passed as a parameter to a method that expect an object of the corresponding wrapper class.	2.	It applies when an object of a wrapper class is passed as a parameter to a method that expects a value of the corresponding primitive type.

. **LOOP**

1.What is loop? Name three types of loop.

It is a iterative structure which performs same work repeatedly based on condition

i) for loop, ii) while loop iii)do while loop

2. what is null loop?

The loop which has no body is known as null loop. If we put semicolon after loop then it becomes null loop.

int i;

for(int i=1;i<=10;i++);

s=s+i;

sopln("sum="+s);

3.Define infinite loop

The loop which never ends due to improper condition or updation of loop counter then it is known infinite loop.

e.g for(int i=1; i>=1;i++)

sop(i);

4. What is delay loop?

The loop which executes only for consuming timed.. no purposeful work is there. Is known as delay loop.

e.g

int i=1;

while(i<=10){

i++;

}

5. Name of two jump statement.

break and continue

break	**continue**
It terminates inner most loop immediately	It skips the rest part of current iteration and moves the cursor to next iteration

e.g
for(int i=1;i<=5;i++)
{
If(i==3)
break;
sop(i);
}
o/p 1 2
	e.g
	for(int i=1;i<=5;i++)
	{
	If(i==3)
	continue
	sop(i);
	}
	o/p 1 2 4 5

while	**do while**
i)it is entry control loop	i)it is exit control loop
ii)If condition is not satisfied then will not execute	ii)If condition is not satisfied it will execute at least once

e.g
int i=5;
while(i>6)
{
Sop(i);
i++;
}
o/p no output
	e.g
	int i=5;
	do
	{
	Sop(i);
	i++;
	}while(i>6);
	o/p 5

for	while
Here initialization, checking and updation are at same place	Here initialization is above, checking is at the starting of the loop, updation is inside the loop
Number of iteration is known in advance	Number of iteration is not known in advance

e.g for(int i=1;i<=10;I++)
s=s+i
	Int i=1;
	While(I<=10)
	{
	S=s+i;
	i=i+1;
	}

Math.max(34,67.3)=67.3

Math.max(7.8,9)=9.0

Math.min(8,9)=8

Math.min(5,7.8)=5.0

Math.sqrt(25)=5.0

Math.pow(4,2)=16.0

Math.ceil(4.5)=5.0

Math.ceil(6.7)=7.0

Math.ceil(-7.8)=-7.0

Math.floor(5.61)=5.0

Math.floor(-6.7)=-7.0

Math.abs(-6.7)=6.7

Math.abs(5.6)=5.6

Math.abs(-7)=7

Math.cbrt(8)=2.0

Math.round(3.4)=3

Math.round(3.5)=4

Math.round(3.7)=4

//Math.round(-4.3)=-4

//Math.round(-4.5)=-4

//Math.round(-4.8)=-5

Math.rint(3.4)=3.0

Math.rint(3.5)=3.0

Math.rint(3.51)=4.0

Math.rint(3.7)=4.0

Math.random()//it gives any value in between 0 to 1

Q)Math.sqrt(Math.floor(16.5))

Math.sqrt(16.0)=4.0

Q)Math.abs(Math.ceil(-4.6))

Math.abs(-4.0)

=4.0

Return type double->

ceil,floor,rint,pow,sqrt,random,cbrt

int->round

any kind->max, min, abs,

Math.ceil(5)=5.0

Math.floor(-6)=-6.0

c.w copy

1)Math.ceil(-9.6)=-9.0

2)Math.sqrt(25)=5.0

3)Math.pow(3,4)=3^4=81.0

4)Math.round(6.7)=7

5)Math.rint(6.5)=6.0

6)Math.abs(-4)=4

7)Math.ceil(Math.abs(-4))

Math.ceil(4)

8)Math.sqrt(Math.abs(Math.floor(-15.7))

chapter 1(oop object oriented program)

1.Difference between oop and pop(procedure oriented program)

oop gives more stress on data rather than method hence any change within the data the method will be automatically changed.

But in case of pop it gives more stress on method rather than data .Here method have to change manually when data is changed.

2.Feature of OOP

i)Emphasis on data

ii)divides the program into some modules

iii)data is hidden and is more secured

3.Basic elements or principal of oop

i)Encapsulation(wrapping up of data member and method together)

ii)Abstraction(represents essential feature without including background details)

iii)Inheritance(reusability)

iv)polymorphism(method overloading)

v)class

vi)object

4.what is class?

A class represents a set of objects having similar characteristics and behaviour where characteristics are represented through data and behaviour are represented through method

5. what is object?

An object is an instance of class having same characteristics and behaviour where characteristics are represented through data and behaviour are represented through method.It is unique entity of a class.

e.g leaving half page

6. what is encapsulation?

It is an oop feature which encapsulates data member and member method together into a single unit called class.

7.what is abstraction?

It is an oop feature which represents essential feature without including background details.

e.g leaving half page

8.what is inheritance?

It is the capability of one class to inherit data member and member method of another class according access specifier. Here the class from which properties are inherited is known as super class and the class which inherits properties are known as subclass.

leaving half page

9. what is polymorphism?

It is a process of using a fuction for more than one purposes. It allows the use of different internal structure by keeping same external interface.

e.g method overloading, method overriding

10.why is class called object factory?

A class is a blue print of objects. It can represent a set of objects having similar characteristics and behavior where characteristics are represented through data and behavior are represented through method. Like a factory can create so many things of same type class also can represent so many object of same type.That is why class is called object factory.

11.Why is object called instance of class?

An object is a unique entity of class . It has same characteristics and behavior of the the class where characteristics are represented through data and behavior are represented through method. That is why object is called instance of class

Leaving half page

12.why is class called user defined data type?

Primitive data are not sufficient to represent every real object. Hence user needs to create data type with the help of class as per his requirement .hence class is called user defined data type.

Leaving half page

13.why is class called user composite data type?

A class can contain primitive as well as non primitive data as its data member. Hence class is called composite data

Leaving half page

14.Difference between class and object

Class	object
Blue print of object	it is an unique entity of class having same characteristics and behavior
It is an object factory	It is an instance of class
No memory space is created for class	Memory space is created for object

Test 1: 4/7/2020 10*2=20

1.write features of java

2.write 4 elements of oop

3.define encapsulation

4.define abstraction

5.what is inheritance? Define super class and subclass

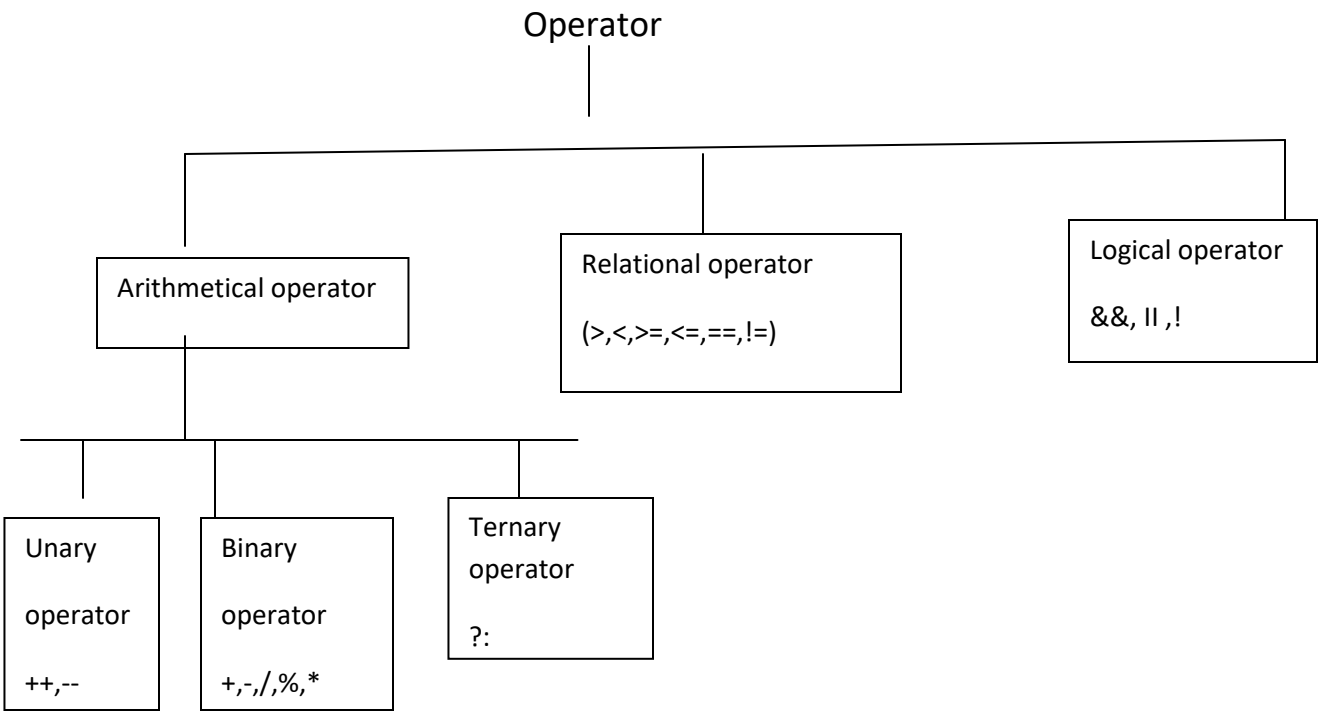
6.difference between class and object

7.why is class called object factory?

8.why is object called instance of class?

9.why is class called composite data type?

10 example of polymorphism



1.Define operator?

Operator is basically a symbol or token which performs arithmetical and logical operations and gives a meaningful result

2.Define operand?

The values which are involved in the operation are termed as operand;

e.g..z=a+b*c

a,b,c are operand here

3. what is expression?

It is the combination of operator , operand and constant values to yield a meaningful result.

int z=a+2*b+6;

4.what is conditional operator?

It is ternary operator(?:) which works on three operand. A relational operator works here if it is true then the value after '?' returns otherwise the value after' :'

returns.

Syntax:value=(condition)?true value:false value

e.g max=(a>b)?a:b

5.what is precedence of operator?

It is the order by which operators are evaluated in an expression . If more than one has same precedence then they are executed as per their associativity.

operator
[],()
++ , -- , !
new (type)
*,/,%
+, -
>,<,>=,<=,==,!=
&&
||
?:
=, +=,-=

Unary	**binary**
It works on single operand	It works on two operand
It can increase or decrease value only	It can do any arithmetical operation
e.g ++,--	e.g +,-,*,/,%

Prefix operator	**Postfix operator**
First change the value and use it	First use the value and change it
e.g int a=5; System.out.println(++a); o/p->6	e.g int a=5; System.out.println(a++); o/p-> 5

Increment operator	**Decrement operator**
i)It increases the value of variable by one .	i)It decreases the value of variable by one
ii)It is of two type. Preincrement, postincrement	ii)it is of two type. Predecrement, postdecrement
e.g int a=7; sop(a++);sop(++a);	e.g int a=7; sop(a--);sop(--a);
Relational operator	Logical Operator
It checks the relation between two values	It works between the results yield by two relational operator
e.g >,<,>=,<=,==,!=	e.g !, &&,||
/	**%**
It gives quotient	It gives remainder
It can works on integer as well as floating point	It can works on only integer
e.g 5/2=2, 7/2=3	e.g 5%3=2,17%4=1

=	==
It ia assignment operator	It is relational operator
It assigns a value to a variable	It checks equality between two values
e.g int a=10;	e.g int a=10,b=20 if(a=b){….}
!	!=
i)It is logical operator	i)It is relational operator
ii)It reverses the value yield by relational operator	ii) it checks inequality between two primitive values
e.g int a=4,b=4 if(!(a==b))	e.g int a=4,b=4; if(a!=b)

STRING

Character class

Method name	Return type	Purpose	example
1.Character.isUpperCase(char)	boolean	It checks whether the given character is in upper case or not	Character.isUpperCase('A') true Character.isUpperCase('a') false Character.isUpperCase('?') false
2.Character.isLowerCase(char)	boolean	It checks whether the given character is in lower case or not	Character.isLowerCase('A') false Character.isLowerCase('a') true Character.isLowerCase('?') false
3.Character.toUpperCase(char)	char	It converts the given character to uppercase	Character.toUpperCase('A') A Character.toUpperCase('a') A Character.toUpperCase('?') ?
4.Character.toLowerCase(char)	char	It converts the given character to lowercase	Character.toLowerCase('A') a Character.toLowerCase('a') a Character.toLowerCase('?') ?
5.Character.isLetter(char)	boolean	It checks whether the given character is a letter or not	Chacter.isLetter('a')true Character.isLetter('1')false Character.isLetter('?')false
6.Character.isDigit(char)	boolean	It checks whether the given character is a digit or not	Chacter.isDigit('a')false Character.isDigit('1')true Character.isDigit('!')false
7.Character.isLetterOrDigit(char)	boolean	It checks whether the given character is a digit or letter	Chacter.isLetterOrDigit('a')true Character.isLetterOrDigit('1')true
8.Character.isWhiteSpace(char)	boolean	It checks whether the given character is a space or not	Character.isWhiteSpace(' ')true Character.isWhiteSpace('A')false

Return type	Method name	Purpose	example
9.int	length()	It returns the number of character in the string	“KOLKATA”.length()->7 “BLUE SKY”.length()->8
10.char	charAt(int)	It returns the character in the specified index	“KOLKATA”.charAt(2)->L
11.int	indexOf(char)	It returns the index (first occurance)of specified character	“KOLKATA”.indexOf(‘K’)->0 “KOLKATA”.indexOf(‘a’)->-1 “KOLKATA”.indexOf(‘p’)->-1
12.int	lastIndexOf(char)	It returns the index (last occurance) of specified character	“KOLKATA”.lastIndexOf(‘K’)->3 “KOLKATA”.lastIndexOf(‘p’)-> -1
13.int	indexOf(char,int)	It returns the index(first occurance)of specified character starting from specified index	“MALAYALAM”.indexOf(‘A’,2)->3 “MALAYALAM”.indexOf(‘A’,5)->5
14.boolean	equals(string)	It checks the equality between two string values considering case	“KOLKATA”.equals(“KOLKATA”)->true “KOLKATA”.equals(“Kolkata”)->false “KOLKATA”.equals(“PUNE”)->false
15.boolean	equalsIgnoreCase(String)	It checks equality between two string ignoring case	“KOLKATA”.equalsIgnoreCase(“KOLKATA”)->true “KOLKATA”.equalsIgnoreCase(“Kolkata”)->true “KOLKATA”.equalsIgnoreCase(“PUNE”)->false
16.String	replace(char,char)	It replaces first specified character by the last specified character in the string	“MALAYALAM”.replace(‘M’,’P’)->PALAYALAP
17.String	replace(String,String)	It replaces first specified String by the last specified String in the string	“green parrot on a green tree”.replace(“green”, ”red”)->red parrot on a red tree
18.int	compareTo(String)	It compares two String lexicographically	“RAM”.compareTo(“RAHIM”)-> 5 “RAHIM”.compareTo(“RAM”)-> -5 “SUN”.compareTo(“SUNNY”)-> -2 “SUNNY”.compareTo(“SUN”)-> 2 “SUN”.compareTo(“SUN”)-> 0 “well’.compareTo(“Welcome”)-> 32 “HAT”.compareTo(“bat”)-> -26
19.int	compareToIgnoreCase(String)	It compares two String lexicographically ignoring case	RAM”.compareToIgnoreCase(“RAHIM”)-> 5 “rahim”.compareToIgnoreCase(“RAM”)-> -5 “SUN”.compareToIgnoreCase(“SUNNY”)->-2 “SUNNY”.compareToIgnoreCase(“SUN”)->2 “HAT”.compareToIgnoreCase(“bat”)=6

			“SUN”.compareToIgnoreCase(“SUN”)->0 “well’.compareToIgnoreCase(“Welcome”)->9 “HAT”.compareToIgnoreCase(“bat”)->6

Return type	Method name	purpose	example
20.String	toUpperCase(String)	Converts the total string to uppercase	“Apple”.toUpperCase()->APPLE “APPLE”.toUpperCase()->APPLE
21.String	toLowerCase(String)	Converts the total string to lowercase	Apple”.toLowerCase()->apple “APPLE”.toLowerCase()->apple
22.String	concat(String)	It concats two string	“Kolkata”.concat(”pune”)->Kolkatapune
23.String	substring(int)	It returns group of characters from the specified index to the end of the string	“Kolkata”.substring(3)->kata
24.String	substring(int,int)	It returns group of characters from the first specified index before the last specified index of the string	“Kolkata”.substring(2,5)->lka
25.int	Integer.parseInt(string) Or Integer.valueOf(String)	Converts string to primitive int	Integer.parseInt(“234”)->234
26.double	Double.parseDouble(String) Or Double.valueOf(String)	Converts string to primitive double	String st=”67.8”; Double.parseDouble(st);->67.8
27.String	Integer.toString(int)	It converts int to String	String st=Integer.toString(45) Sop(st)->45
28.String	Double.toString(double)	It converts double to String	double d=7.8 String s=Double.toString(d); Sop(s);->7.8
29.String	String.valueOf(char)	Here character is converted to String	String.valueOf(‘a’)=”a”
30.String	trim()	It removes the leading and trailing spaces	“ Kolkata city “ Kolkata city
31.boolean	startsWith(String)	It checks the given string starts with the parameterized string or not	“KOLKATA”.startsWith(“KOL”)->true “KOLKATA”.startsWith(“K”)->true “KOLKATA”.startsWith(“kol”)->false
32.boolean	endsWith(String)	It checks the given string ends with the parameterized string or not	“KOLKATA”.endsWith(“KATA”)->true “KOLKATA”endsWith(“A”)->true “KOLKATA”endsWith(“kol”)->false

VALUES AND DATATYPE

1.what is token?

Each smallest individual unit used in java program is known as token. E.g

Literal,operator, separator,punctuator ,data type etc

2.what is literal?

Each constant value used in java program is known as literal.

e.g

int literal(5,7,8,3)

double literal(5.6,7.8)

char literal ('a','d')

3.what is variable?

A variable is a named memory location which contains a value.

e.g int a=10;

'a' is a variable

4.what is identifier?

It is an user given name to class array method etc.

e.g Scanner sc=new Scanner(System.in); where 'sc' is a Scanner object

5.what is data type?

Data type is the type or kind of data and associated operation to handle them.

Two type:

Primitive data and non primitive data

6. Dfference between primitive and non primitive data

Primitive data	Non primitive data
i)fundamental data	i) derived data
ii)independent data	ii)indirectly dependent on primitive data

Iii) they have fixed length	iii)their length depends on user
e.g int , float, char etc	e.g class, array etc

7.what type conversion and what are different kind of type conversion?

In a mixed expression when data are converted to a single data type then it is known as type conversion.

Two type of conversion.

i)implicite type conversion ii)explicite type conversion

8.Difference between implicite and explicite type conversion

Implicite type conversion/coercion	Explicite conversion/casting
i)Here data is converted to higher type	i) Here data is converted to desired type
ii)User intervension is not needed here	ii)User intervension is needed here
iii)data value will not loss here	iii)data value may or may not loss here

e.g int a=8; double d=7.8
d=a;
	e.g int a=8; double d=7.8
	a=(int)d;
	d=(double)a;

9.what is escape sequence?

There are some non graphic character which are used as command to direct the cursor while printing.

e.g \t horizoantal tab

\n new line

10.what is punctuator?

There are some punctuation sign used as special character in java. E.g '?', ':' ,'.'

11.what is separator?

Separator are used to separate same kind of variable . ',' is used as separator.

12. Write down naming rules for variable

i)A variable may have any number of character.

ii)It may contain alphabets, digits and underscore

iii)Variable name should be meaningful

iv)special character '_' can be applied in between any two characters

13. Reasons for a variable to be invalid?

i) A variable name should not starts with digit.

ii)A variable name should not include any space in between two characters

iii)It should not be any keyword

14. What is arithmetic expression and what are different kind of arithmetic expression ?

Give definition and example

An expression which contains variable constants and arithmetic operators is termed as arithmetic expression

Two type:

i)Pure Arithmatic Expression: In this expression it contains same type of data items(variable and constant) e.g int a=10; int x=a*2+20;

ii)Impure Arithmatic Expression: In this expression it contains different type of data items(variable and constant) e.g double a=5.6; int b=8; double x=a+b*2;

15.Description about primitive data

Data type	Size	Default value	range
byte	1 byte(8 bits)	0	-128 to +127
boolean	1 byte	false	true, false
char	2 byte(16 bits)	'\u0000'	0 to 65535
short	2 byte(16 bits)	0	-32768 to 32767
int	4 byte(32 bits)	0	-2^31 to 2^(31)-1
float	4 byte(32 bits)	0.0f	
long	8 byte(64 bits)	0 L	-2^63 to 2^(63)-1
double	8 byte(64 bits)	0.0 d	

16.List of escape sequence

\t->horizontal tab	\'->single quote
\v->vertical tab	\''->double quote
\\->back slash	\b->back space

\0->null	\n->new line

Here's the full list:

- \t - tab.
- \b - backspace (a step backward in the text or deletion of a single character).
- \n - new line.
- \r - carriage return. ()
- \f - form feed.
- \' single quote.
- \" double quote.
- \\ **backslash**.

Satic initialization	Dianamic initialization
Value to the variable is given at compile time	Value to the variable is given at run time
e.g int a=10;	e.g int a=sc.nextInt(); where sc ia a Scanner object

Char	**String**
Take one character	**Takes group of charactres**
Enclosed within single quote	**Enclosed within double quote**
e.g char ch='a'	**String st="a", st1="Kolkata"**
float	**double**
It has memory space 4 byte	**It has memory space 8 byte**
It takes decimal place upto 4 place	**It take decimal places upto 8 place**
Dedfault value 0.0 f	**Default value 0.0d**

Int	**float**
Takes only integer value	**Takes integer as well as floating value**
Default value 0	**Default value 0.0f**

Wrapper class: it is a member of java library java.lang which converts primitive data in terms of object. it also provides conversion

Primitivedata value	wrapper class	size(byte)	default
byte	Byte	1	0
boolean	Boolean	1	false
short	Short	2	0
char	Character	2	'\u0000'
int	Integer	4	0
float	Float	4	0.0f/0.0F
long	Long	8	0L
double	Double	8	0.0d

Wrapper class method description:

1)

a)Integer ob=new Integer(5);//autoboxing(converting primitive value to object)

Integer obj=5;//autoboxing

b)int a=ob;//unboxing(converting object to primitive value)

OR

Int a=ob.intValue()//(converting object to primitive value)

2)

a)int a=Integer.parseInt(“5”);(conversion from string to int)

OR

int a=Integer.valueOf(“5”);(converting string to int)

b)String s=Integer.toString(100);(conversion from int to string)

c)double d=Double.parseDouble(“8.9”)-> (converting String to double)

OR

double d=Double.valueOf(“8.9”) -> (converting String to double)

d)Double ob=new Double(7.8);//primitive data to object

3)

static methods is called directly with class name

all methods(e.g parseDouble(), toString(), valueOf())of wrapper

4)e.g

String a=”20”,b=”30”;

Sopln(a+b);

int c=Integer.parseInt(a);

int d=Integer.parseInt(b);

Sopln(c+d);

o/p

2030

50

//package contains group of classes

Package java.util for Scanner class

Package java.io for BufferedReader class

anything else->package java.lang

default package(java.lang)