UNIT -1 : JAVA BASICS

Q-1.history of java OR background of java

ANS:-

- Java was developed by sun microsystem in 1991.
- In early year java was known as OAK programming language.
- Now OAK was renamed by java in 1994.
- Java was released in may 27,1995.
- Java is mainly developed for internet based application.
- Java provides a facility to develop different types of application like:
  - Desktop application
  - Mobile application
  - Web based application
- The java is fully object oriented programming language.
- The father of the java is james gosling.

Q-2.what is java? Why it is known as platform independent language.
OR Explain byte code.

ANS:-

- The java is oop language because java program stored into small parts called "object".
- The main purpose of java is to create a computer language which can we used by different execution environment.
- Java support variety of hardware and software environment .
- Java has ability to write portable program and execute on different operating system.
- There are 3 types of application created using java.
  - Desktop application
  - Mobile application
  - Enterprise application(distributed application)
- many other programming language does not execute on different platform.
- Java generated byte code and java applications executed on different platform so java is known as platform independent language.

❖ **BYTE CODE:-**

ByteCode is code that Executable on any platform Environment.

- When java program is complied that time it generated .class file.
- This class file has byte code.
- Byte code is created by JVM(JAVA VERTUAL MACHINE) and that is easily executed an different platforms so byte code is capable to transfer java program from one cpu to another cpu so java is platform independent language.

Q-3.explain basic type of java.

ANS:-

- Java provides basic 8 data type.
- Data type is used to specify which type of value stored in variable.

1).CHARACTER:-

- It is used to store character value(one character).
- It allocates two bytes.
  - Key word:-CHAR
  - Example:-char nm=a;

2).BOOLEAN:-

- This data type is used to store Boolean type of value like:-true or false , yes or no.

- It allocate 1 bit.
    - Key word:-boolean
    - Example:-boolean n=true;

3).BYTE:-

- It is used to store numerical value without fractional (without decimal point).
- It allocates one byte.
    - Key word:-byte
    - Example: -128 to +127 range
      Byte b=100;
- The range is -128 to 127.

4).SHORT:-

- This data type is used to store intigeral value.
- It allocated two bytes.
    - Key word:-sort
    - Example:-short b=120;
- The range of short -32768 to 32767.

5).INT:-

- This data type is also to store integer value but it is larger than short integer.
- It allocates four bytes.
    - Key word:-int
    - Example:-int 125;

6).LONG:-

- It is used to store integeral value larger than integer data type.
- It allocated eight byte.
    - Key word:-long
    - Example:-long 1=150;

7).FLOAT:-

- This data type is used to store fractional value (with point).
- It allocated four byte.
    - Key word:-float
    - Example:-float pi=3.14;

8).DOUBLE:-

- This data type is also used to store floating value larger than float data type
- It allocates eight byte.
    - Key word:-double
    - Example:-double a=180;

DATA TYPE	***USE***	***KEYWORD***	**MEMORY Occu.**
1).boolean	True/false	boolean	1 bit
2).character	One char	char	2 byte
3).byte	-128 to 127	byte	1 byte
4).short	-32768 to 32767	short	2 byte
5).int	Store int value	int	4 byte
6).long	Store long value	long	8 byte
7).float	Store float value	float	4 byte
8).double	Store double value	double	8 byte

Q-4 explain characteristics of java.

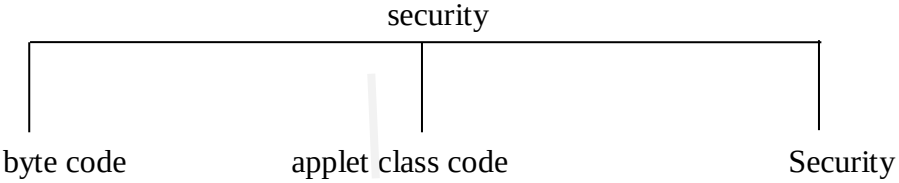
Ans. Java provide many different type of characteristics.

1. Simple:-

- If user know basic concept of oop then java will be easier to learn.
- Java is designed to be easy for the progrsmmer.
- Java provides set of class library to build application.

2. Secure:-

- Java does not allow to access memory directly because java does not Support pointer.



3.Portable:-

- Java ensure that other implementation of language specification that affect in current version.
- Java program can be easily transfer to any platform because it generate byte code so java is portable. (file size of primitive data types)

4.object oriented:-

- Java is fully ooplanguage(collection of objects).
- In java all the data placed within a class and accessed using object.So , java is object oriented language.

5.robust:-

- Java is capable for providing powerful functionality to build different application.
- It also provides exception handling and managing runtime error.
- It provides dynamic memory allocation & memory management.
- So java is robust language because it has start compile time checking error runtime.

6.multi threaded:-

- It is a process of sharing resources by multiple users can perform many task simultaneously.
- Java provides multi threading to build network application.

7.high performance:-

- Java generate byte code which is executable on any platform so java provides high performance.

8.distributed:-

- Java is capable to develop internet based application(web application).
- User can build client-server based Application using java.

9.dynamic:-

- It is more dynamic than it can any extensive amount of runtime information.

Q-5.write a note on array.

ANS:

- Array is a collection of more than one values in a single variable.
- Array element is accessed from its position called index.
- Index start from 0.

ARRAY PROPERTY:-

1.length:-

- This property returns the size of array.
- Java provides 3 types of array.

1. **Single dimensional array:-**

- When are is declared by using only row size it is known as single dimensional array.
  - **SYNTAX:-**
    - Datatypevarname=new type[size];
    - Int [] rno=new int[5];

  Memory Structure:

11	22	33	44	55
0	1	2	3	4 (index)

2. **Multi dimensional array:-**

- When array is declare with row and column size than it is known as multi dimensional array.
- In this type of array each row contains fixed column.
  - **SYNTAX:-**
    - Data type [][] variable=new type[row][column];
    - Int [][] a=new int[2][4];

Memory Structure:

3. **Jagged array:-**

- In jagged array the size of the column are not fixed.
- When there is a fixed size of row but each row contains different size of columns than it is known as jagged array.
  - **SYNTAX:-**
    - Int stud[][]=new int[4][];
- After declaring row size define the column size for each row.
  - Stud[0]=new int[2];
  - Stud[1]=new int[4];
  - Stud[2]=new int[3];
  - Stud[3]=new int[2];

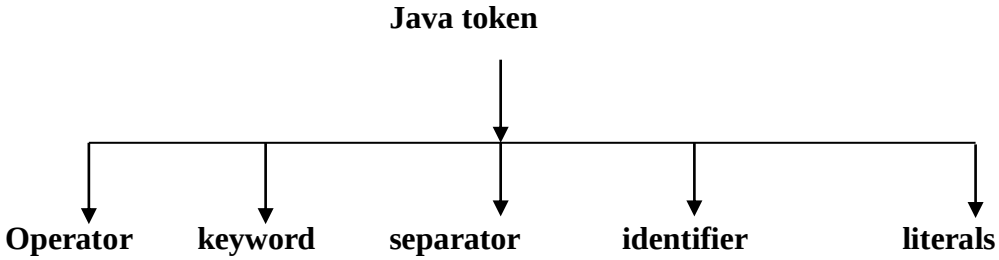
Memory Structure:

00	01		
10	11	12	13
20	21	22	
30		31	

Q-6.explain java tokens.

ANS.

- Tokens are smallest unit used to built java program.
- Java token are divided into 5 catagries.



1.operator:

- It is a graphical symbol used to perform specific task.
- Java provides many types of operator.
  - Arithmetic operator
  - Logical operator
  - Conditional operator
  - Relational operator
  - Bit wise operator
  - Special operator

2. key word:-

- The reserve word of java is known as key word.
- Java provide like

  if, else, do, while, public

3. separator:-

- Separator are used to inform compiler how data is divided or code is depart from each other.

SEPARATOR	USE
;	Used to terminate line
,	Used to declare multiple variable
{}	Used to define body of method and class
.	Used to access method of class using object
[]	Used to define array

4.identifier:-

- It is the unique name to identify class, variable and method.
- Identifier must be valid like
  - Upper case(A to Z)
  - Lower case(a to z)
  - Digit(0 to 9)
  - Symbol(-)

5.literals:-

- Literals comment and white space in program
  1. Single line comment(//)
  2. Multi line comment(/* */)

Q-7.explain operator in java

ANS:-

- Operator is a graphical symbol used to perform specific task.
- Java provides 6 type of operator.

1.arithmetic operator:-

- Used to perform arithmetic or mathematical calculation.

OPERATOR	USE
+	Perform addition
-	Perform subtraction
*	Perform multiplication
/	Perform division
%	Perform modules

2.logical operator:-

- Used to test more than one condition at a time.

OPERATOR	USE
AND(&&)	All condition must be true
OR(||)	Any one condition must be true
NOT(!)	Condition must be false

3.relational operator:-

- It is also known as comparison operator.
- Used to compare 2 variable.

OPERATOR	USE
>	Used to grater than
<	Used to less than
>=	Used to grater than equal to
<=	Used to less than equal to
==	Equal to
!=	Not equal to

4.conditional operator:-

- It is called ternary operator.
- It works like if else.
- It contains 3 argument.

OPERATOR	USE
Condition	It is the logical test
True expression	If condition is true than second part will executed
False expression	If condition is false than third part will be execute

SYNTAX:-

Condition?<true exp>:<false exp>

5.assignment operator:-

- It is used to assigned value into variable.

OPERATOR	USE
=	It assign a value
+=	It adds and assign
-=	It subtraction and assign
*=	It multiplication and assign
/=	It division and assign

6.bitwise operator:-

- It work on bits.
- It perform bit by bit operation and uses value.

OPERATOR	USE
&(binary AND)	If both operand contains 1 bit than result will be 1 otherwise0
||(binary OR)	If any one operand contains 1 bit than result will be 1 otherwise 0
^(binary XOR)	If only and only one operand contains 1 bit than result will be 1 otherwise 0

~(once complement)	It is use to flip bits 1 replaced by 0 and 0 replaced by 1
<<(left shift operator)	Used to move no. of bits left side
>>(right shift operator)	Used to move no of bits right side

7.special operator:-

- Java provides some special type of operator.

OPERATOR	USE
.(dot)	It is use to access methods and variables of class using object
+(concatenation)	Used to concate string and any other variable
instanse of	It is used to check whether the object is any class type or not

Q-8. Explain commend line argument.

ANS.

- in a commend line arguments all the arguments are passed from commend line/commend prompt.
- In this process all the arguments are stored in string array of main function each argument is separated by space.
- User can pass no of argument of any datatype.
- All the different argument are converted into string type.
- So user passes arguments using commend prompt is known commend line argument.
  - **SYNTAX:-**
    - Java program arg1,arg2,……,argn

```
Class sy
{
        public static void main(String []ar)
        {
                System.out.println(ar.length);
        }
}
>javac sy.java
>java sy welcome to clg
O/p : 3
```

Q-9 explain JDK tool <u>OR</u> JDK components.

ANS.

- JDK stands for java development kit it is a installation package java.
- It provides many components.

JDK tool	Meaning	Use
1.javac	Java compiler	Used to compile source file into byte code
2.java	Java interpreter	Used to execute java application
3.javadoc	Java document	Provides documentions help file
4.JDB	Java debugger	Used to debug the program
5.javap	Java desembler	Used to convert byte code into source program
6.applet viewer	-	Used to executed applet

Q-10. Explain public static void main(String [] args).

ANS.

- main() is the starting execution point for compiler.
- Each program contains only one main().
  - **SYNTAX:-**
    - Public static void main(String ar[])
      {
      //block of code;
      }

1. Public:-

- It is access modifier.
- Main() is executed from in class so it is public.

2. Static:-

- Static determined the main() is called without creating instance(object).

3. Void:-

- Void determined does not return any value.

4. Main():-

- It is the entry point or string point of program for execution.

5. (String ar[]):-

- It is the string type array.
- It is used to store value passed from commend line.

Q-11 explain java Environment. OR explain JVM.

ANS.

- Java environment provides 3 of component.
  1. JDK
  2. JRE
  3. JVM

1. JDK(java development kit):-

- JDK is a installation package of java.
- JDK provides all the components and tools build java application.
- It provide set of class library.

2. JRE(java runtime environment):-

- JRE is used run the or execute the java program.
- JRE extends the byte code.
- JRE is also manage the memory and handle the exception.

3. JVM(java virtual machine):-

- JVM is the java compiler.
- JVM compiles the java program and converted into byte code.

Part-2 Class & Object

Q.1 What is Class & Object.

Class

- Class is a group of variables and methods.
- The class is a main component of oop language.
- Class allow user to bind variable and method in a single unit or group.
- So class define common template for design for object.
- The property of class is called member variable (data member) and member function.

Syntax:-

```
Class classname
{
        //Data member
        //member function
}
```

- Class is a keyword used to define class within a class you can define variables and methods.

 Object

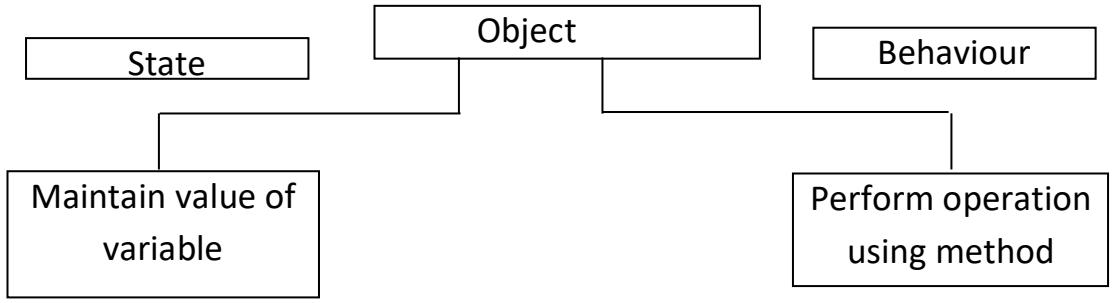
- an object is a Runtime Entity.
- an object is region storage area.
- when instance of class is created is known as object.
- Class variable and methods are accessible outside of class using object.

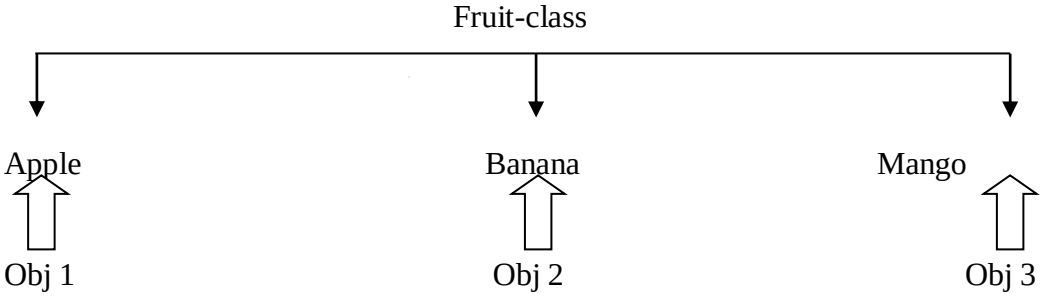
Syntax:-

```
Classnameobjectname=new classname();
Objectname.methodname();
```

- new keyword is used to allocate the memory for the object.
- object has to task

1) it store and maintain the value of variable.
2) it performs the operation using method.





Q.2 Method Overloading

- if there are more than one method of same name define in a single class than it is called method overloading.

- in a method overloading class contain same method more than one time.
- in method overloading must not be duplicate.

 Characteristic/Rules:-

(1) Class must contain more than one method of same name.
(2) Number of argument must be different.
(3) Data type of argument must be different.
(4) Return type of method should be different.

Example:-

```
Void sum()
{
}
Void sum(int i)
{
}
Void sum(inti,int j)
{
}
int sum(String s)
{
}
```

Q.3 what is constructor? Explain type of constructor.

Or

Explain constructor overloading.

- Constructor is a special member function which has same as class name.
- When object of that class is created that time constructor is automatic executed.
- Constructor is used to initialize resources when object is created.likevariable,database connection open,file open etc.

 Characteristics:-

1) It has same as class name.
2) It is automatic executed when object is created.
3) Do not need to call constructor.
4) Constructor do not have any return type.
5) It is used to assign value of variable (instance variable) (object)

 Constructor overloading:-

- When there are more than one constructors are define in a single class than it is known as constructor overloading.
- There are two types of constructor

1) Default constructor
2) Parameterized constructor

1) Default constructor:-

- When there is no argument in constructor than it is called default constructor.

2) Parameterized constructor:-

- When constructor contains any type of argument than it is called parameterized constructor.

Example:

```
Class sy
{
        Sy()
        {
            //default Constructor        }
```

```
        Sy (intI,int j)
        {
            //parameterized constructor
        }
}
```

Q.4 Static keyword:-

- Static is a keyword it is not associated with object.
- Static keyword referenced to class directly.
- There are three ways to static
  1) Static variable
  2) Static method
  3) Static block

1) Static variable:-

- Static variable is shared by all object of class.
- Instance variable have(class variable)separate memory for storage but there is need to be common variable between all objects than create static variable.
- Static variable is initialized only one when class is loaded into memory.
- Static variable can be accessed directly by classname.

Syntax:-

```
Static datatype variable=value;
```

2) Static method:-

- When method is define using static keyword than it is called static method.
- Static method is executed directly by the classname do not need to create object.
- Static method can only access only static variable.
- It can not access non static variable.

Syntax:-

```
Static void main()
{
    //BOC
}
```

3) static block:-

- The static block is a block of statement inside a class.
- Static block is used to initialize static variable.
- It is executed only once when class is loaded into memory.

Syntax:-

```
Static
{
    //BOC
}
```

Q.5 Explain varargs.

- Varargs is known as variable arguments(varing arguments).
- Normal method can take specific number of(fix number) arguments so user can not pass more than or less than specific arguments.
- Varargs provides a facility to pass different member of arguments while calling.

Syntax:-

```
return type <function name>     (datatype...varname)
{
    //BOC                   }
```

- Number of argument are stored in variable so it becomes array variable.

Example:

```
Void sum(int… no)
{
        System.out.println(no.length);
}
Sum(10);
Sum(10,20,30,40,50);
```

Q.6 Explain finalize() method of java.

Or Explain garbage collection in java.

- It is a special method of java.
- Finalize method () is used to free or remove the memory and resources.
- Java provides a facility to free the memory of object it is known as garbage collection.
- Garbage collection is automatically done by JVM.

  Syntax:-

```
Void finalize()
{
        //BOC
}
```

- Finalize method provides a facility for any object to destroy the memory.
- Finalize () is used
  1) Release the resource.
  2) Free the memory.
  3) Close the collection with database.
  4) Close the file.
- Finalize dose not return a value.

UNIT – 2 : INHERIRANCE & JAVA PACKAGES

Q.1 What is inheritance or explain types of inheritance?

- the property of exiting of class is used into new class than it is known as inheritance.
- inheritance provides **general class (general design)** that defines set of methods.
- In inheritance mechanism exiting class is known as base class or super class.
- The new class is known as derive class, sub class or child class.

Syntax: -

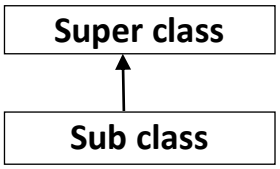
```
Class <sub class name> extends <base class name>
{
        //BOC
}
```

- Extends keyword is use to inherit the class

❖ Types of inheritance: -

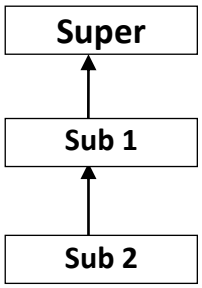
1. Single / Simple inheritance: -

- When there is one base class and only one derive class than it is called single inheritance.



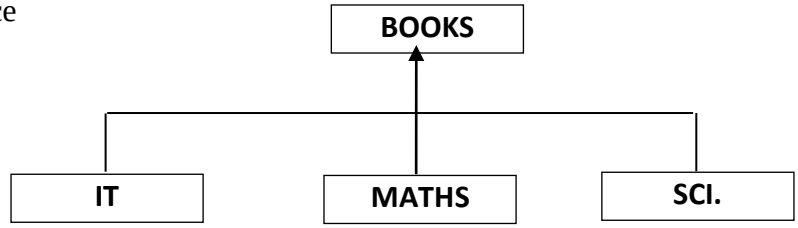
2. Multilevel Inheritance: -

- When any class becomes derive class and also base class of another class than it is called multilevel inheritance.



3. Hierarchical Inheritance: -

- When there is a one base class and more than one derive class than it is called Hierarchical inheritance



Q.2 Explain method overriding?

- When same method is define with same syntax in both class base class or also derive class than t is known as method overriding
-

Super Class

```
Void show()
  {
    a +b;
  }
```

Sub Class

```
Void show ()
   {
     a -b;
   }
```

❖ ***Rules : -***

1) Duplicate method must be define in both class base class also derive class.
   Like : -
   - Return type must be same
   - Method name must be same
   - Argument must be same
2) in method overriding if you want to call method of the base class that time **super** keyword is use
3) final method can not be override
4) private method can not be override
5) static method can not be override
6) constructor can not be override

Q.3Explain super keyword OR How to access constructor in base class(inheritance) ?

- constructor can not be inherited in sub class.
- User can define constructor in both class base class and sub class.
- In java derive class constructor is executed when object is created but base class constructor will not automatically called in derive class.
- Super keyword is use to call the **constructor of base class from derive class**

- **There are two use of super keyword : -**

1. It is used to call base class constructor from the derive constructor in Inheritance.
2. It is use to call method of the base class during overriding.

NOTE :- Super keyword must be a first line into derive class…

supercallingconstructor

```
Base (int)
 {
 }
Void show ()
{   a + b     }
```

Super (i)

Sub Class

```
Derive (int i)
 {
    super(i);
 }
Void show ()
{     super.show ()
       a- b
 }
```

Q.4 Write a note on interface OR How to define multiple interface in java ?

- **Interface is a collection of abstract method.**
- Interface is a same name as class name.
- **Java does not supportmultiple interface** because java do not allow to extend more than one class.
- But user can **inherit multiple interface.**
- Interface only provides **definition of method**it does not have body part.

Syntax: -

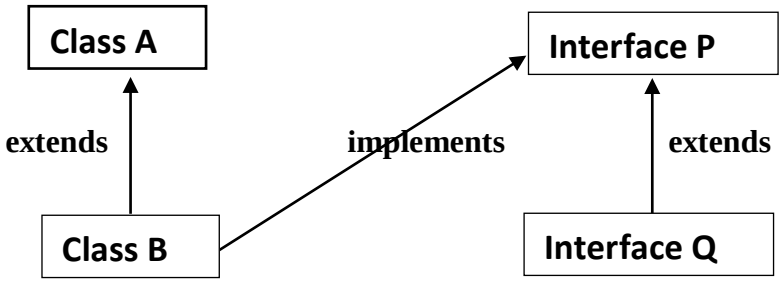
```
interface<interface name>
{
        Void show ();
}
```

- To inherit interface **implement** keyword is used.

```
Class<class name> implements <interface 1,2,…N)
{

}
```

- Interface can extend another **interface using extend keyword.**



❖ **Characteristics of interface :-**

1. Interface provide **multiple interface.**
2. It is a **collection of abstract method.**
3. By default methods are **abstract.**
4. **Keyword : implement.**
5. Can not **create instance of interface.**
6. does**not create constructor.**
7. it does not **create final method.**
8. Methods of interface must be override class.

Q.5 Explain abstract class ?

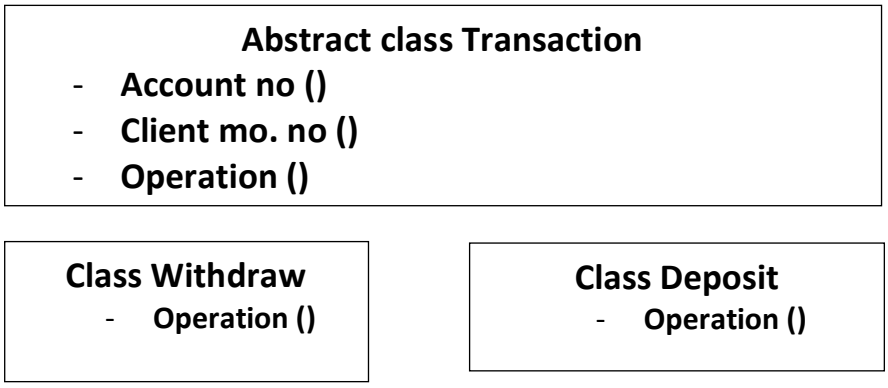
- The class with declared with abstract keyword than it is called abstract class.
- In this class abstract method can be created using abstract keyword.
- Abstract class contains mixture of abstract and non-abstract method.
- When user wont to define some method without body part (abstract method) and some method with code (non-abstract method) than create abstract class.

Note :-

1. Abstract class (0 to 100%) partially.
2. Interface (100%) full abstract.
3. Class (0%) abstract.

Syntax :-

```
abstract class <class name>
{
    //declare abstract methods
  // declare non-abstract method
}
```



Here,withdraw class & Deposit class both can inherit the Abstract Class Transaction.

- **CHARACTORISTIC : -**
  1. Abstract method must define body part in define sub class.
  2. Can not create instance of abstract class.
  3. Abstract method I this class user abstract keyword.

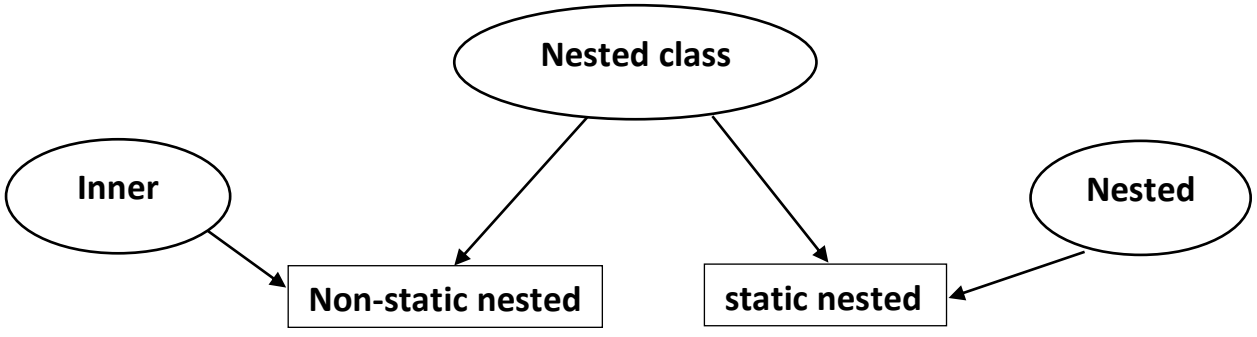
Q.6 Explain nested class in java .
OR
Explain types of nested class .
OR
Explain inner class .

- The class contain another class than it is knows as nested class.

Syntax :-

```
Class outer class
{
    Class nested class
    {
    }
}
```

- There are two types of nested class



1) **Non-static nested class : - (inner class)**

- When nested class is defining**without static keyword than it is known as non-static nested class**
- it is also **called inner class.**
- In this type users have to create instance of inner class into outer class.

Syntax :-

outer o1=new outer();
outer. inside i1=o1.new inside();

2) **Static nested class :- (Nested)**

- When class is define **using static keyword** than it is **called static nested class.**
- in static nested class user do not need to create **object of sub class.**
- It is **directly called by class name.**

Example :-

```
class outer
{
    static class inside
    {
        void print ()
        {
            System.out.println("inside");
        }
    }
    void show()
    {
        System.out.println("outer");
    }
}

class nest
 {
    Public static void main(String ar[])
    {
        outer. inside i1=new outer. inside();
        {
            i.print();
        }
    }
}
```

Ch :-3 Exception handling,Thread& IO Package

Q-1 Write a note on exception handling mechanism.

Or

Explain try and catch block

- Exception is a one type of error which is generated during the execution of program.
- When there is a in logical generated or invalid problem is arise that time exception is generated.
- In exception is arrive due to many reasons like

  a) User entered invalid data.
  b) A file try to open is not exist.
  c) Network communication is lost during process.
  d) User try to divided by 0.
- So, java provides a way to handle exception so it's called exception handling mechanism.

❖ Try and block

Syntax

```
try
{
    //Boc;
}

catch (Exception ex)
{
        //Boc
}
Catch(Exception ex)
{
        //BOC
}
finally
{
        //Boc;
}
```

❖ **Try block**

- Try block is block of statement in which user want to monitoring the code.
- Try blocking process the block if there is any problem then try block generated the exception.
- Each try block must have at list one catch block.

❖ **Catch block**

- Catch is use to handle the exception generated by the try block.
- Catch block put immediately after the try block otherwise compiler generate error.
- User can put more than one catch block but executed only one.
- Catch block catch one argument of exception type.

❖ **FINALLY BLOCK**

- This is the optional block this block each time executed whether the exception is generated or not.
- This block is used to face the resource like data base, connection , file closing in exception.

Q-2 Explain Exception handling keyword

- **Try**
  - This block is used to monitor the code and able to generate exception.
- **Catch**
  - If exception is generated in try block then it is handle by catch block. User can use multiple catch block.

1) Throw

- If user want to manually create and exception that time throw keyword is user

2) Throws

- When any method generates exception that time throws keyword is used.

3) Finally

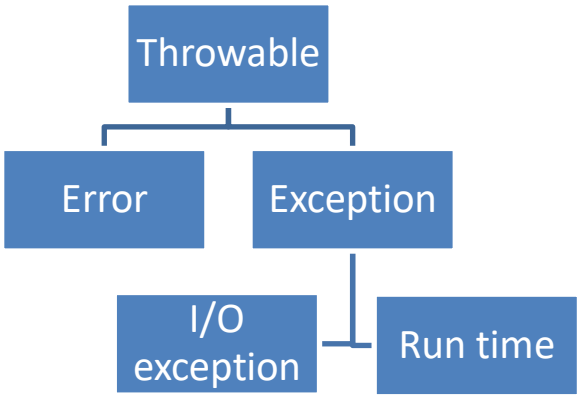
- This block is exception executed each time whether exception is generated is or not

Q-3 Write a note on throwable class.

Or

Explain types of exception.

- Throwable class is a super class of all exception .
- Throwableclasss is place in java.lan.*
- Throwable class is hierarchy



❖ **Error class**

- Error class is used to handle special type of exception like :
  i. Memory overflow exception
  ii. Hardware or software problem etc…
- These types of exception are not handle by user.

❖ **Exception class**

- This class is capable to handle the I/O exception and also run time exception.

a) I/O exception

i. When the any problem related to input or output that time I/O exception is generated

b) Runtime exception

When there is any invalid operation is performed during exception of program that time runtime exception is generated.

There are many different types of exception.

Exception name	Use
ArithmeticException	Divided by zero
ArrayIndexOutOfBoundsException	If user access array index out of rang

NullpointerException	When string is null & try to perform string operation.
NumberFormatException	When user try to convert value into Invalid datatype
NegativeArraysizeException	When user can crate array of negative size

Q-4 Write a note on customize exception

Or

Explain user define exception

Or

Explain throws keyword

- Java has powerful capability to handle the exception.
- Java provides many built in exceptions.
- When user creates its own exception according to user need than it is called customize exception (user define exception).
- In customize exception user creates its own exception class.

❖ Throws keyword
  - When any user defines method is capable to throw the exception that time throws keyword is used.
  - Throws keyword is used by any method but that method can throw the exception not handle the exception.
  - This facility of throwing exception by user makes your application robust.

❖ Syntax:

```
Return type function([ary] throws <exception name>)
{
        //boc
}
```

Q-5 Explain nested try block

Or

Explain throw keyword

- When one try block is placed into another try block than it is known as nested try and block.

❖ Throw keyword
  - When user want to generate exception manually that time throw keyword is used.
  - Throw keyword is also used to forward exception generated by inner try block to outer try block.

❖ Syntax
  - Throw <exception name>;

❖ When you throw the exception manually it must be catch forward otherwise compiler generate error.

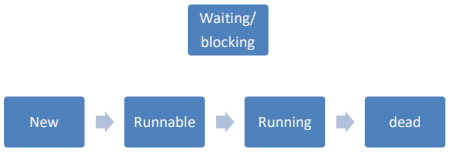
Syntax:

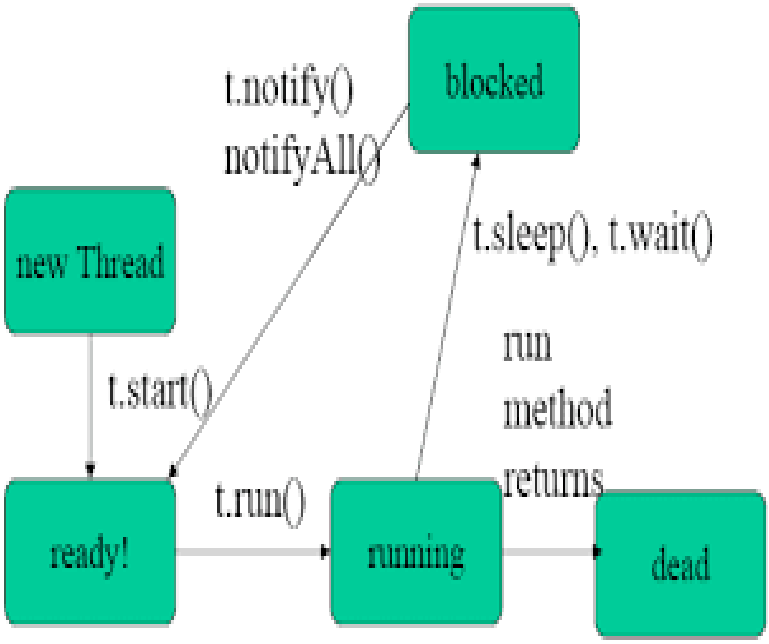
```
try
{
        //block of code
        try
        {
                //block of code
        }
```

```
        Catch(Exception e){}
    }
    Catch(Exception e){}
```

Q-6 what is thread? Explain life cycle of thread.
Or
Explain state of the thread

- A thread is a sequential flow of control in a program.
- Thread is a used to execute the one by one request of client sequentially.
- Thread is a lightweight process used to share same resource between multiple client.
- Thread allows user to access resource according to specific time interval.
- Thread process is divided into 5 categories.





- ❖ Thread life cycle
- ❖ When threads execute the request that time it transfer from one state to another state during execution of process is called life cycle of thread.
- ❖ There are 5 state of thread life cycle.

State	Use
1) New	When state is created but not started
2) Runnable	• Thread immediately transfer from new state to runable state. • It indicate ready for execution
3) Running	It is currently executing thred
4) Waiting / Blocked	When thread transfer from running to delay time than is called waiting / Blocked
5) Terminate / dead	After complication of execution thread transfer to the dead state.

- **There are two way to inherit the thread**
  1) **Thread class :-**
     - Thread class allows user to inherit the thread.

     Syntax
     Public class classname extends thread.
  2) **Runable interface :-**
     - This interface allows user to access thread.

     Syntax
     Public class classname implements runnable.

Q-7 Explain method of thread

- Thread allows user to access one by one process so it also provides many method,

1. **Run()**
   - When thread transfer to running mode that time run method is executed.
   - It contains set of code for execution.
   - It is the abstract method of thread class so user must override in sub class.

   Syntax
   Public void run()
   {
   //Boc
   }
2. **Sleep () :-**
   - This method transfer the thread from running mode to waiting mode from given millisecond.
   - So it is used to delay execution.

   Syntax
   Void sleep (int millisecond)
3. **Start () :-**
   - This method start the execution of thread by calling run method.

   Syntax
   Void start ();
4. **getName () :-**
   - It is used to return name of the thread which is currently executing.

   Syntax
   String getName ();
5. **setName ()** :-
   - It is used to set the name of thread.

   Syntax
   Void setName(String Name)
6. **Suspend () / stop ()/ Yield () :-**
   - This method is used to cancel the execution of thread and transferring from running to blocked / dead mode/

   Syntax
   Void stop ();
7. **getPriority () :-**It returns the Priority number of the thread.
   - **Syntax:** intgetPriority()
8. **setPriority () :-**
   - It is used to set priority number for thread,

   Syntax:
   Void setPriority(int priority);

9. join () :-

- Join method locked all thread until the thread are not completed

Syntax
Void join ()

10. Interrupt()

- Interrupt is also used stop thread before it completed but this method generate interrupted exception.
- This method transfer the thread from running mode to dead mode.
- This method will not effect in all other process.
- Syntax: void Interrupt();

11. isAlive()

- This method checks weather a thread is started or dead.
- If thread is started then this method return true otherwise returns false.
- Syntax: booleanisAlive();

12. join()

- Join waits when thread is completed.
- Whenever join() is used that time it waits untill all threads which are joined are not died.
  Syntax: void join();
  This methods throws the InterruptedException.

Q-8 Synchronization in java
Or
Explain synchronized keyword.

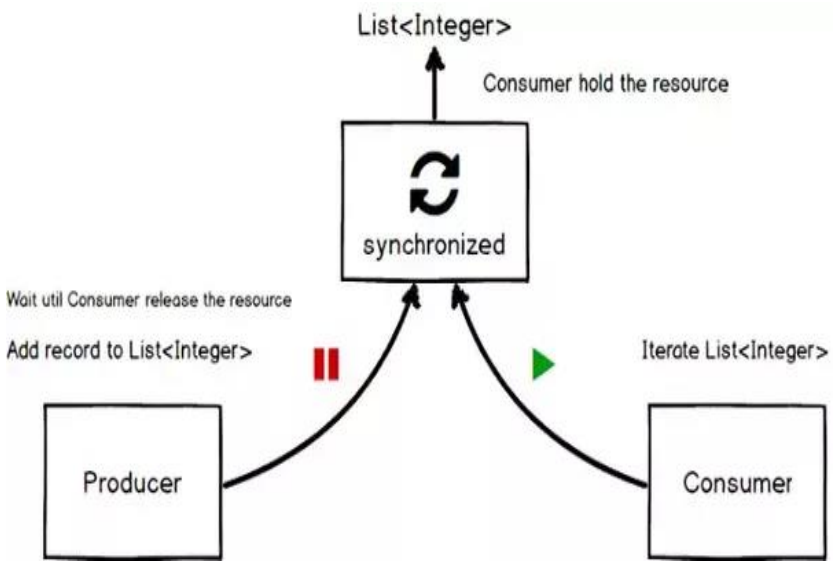
- Synchronization is a process ofmulti threading.
- It allows user to share one resource by multiple user at a time but it allows one by one thread to share resource.
- Synchronization allows only one thread to execute resource at a time all other thread becomes wait.
- Synchronization puts the lock on resource while one thread is executing when thread is completed synchronization realize the lock on resource automatically and then another thread can get the resource.
- So synchronization is a process to monitor the resource and also multiple threads.

Syntax :-

```
Synchronized<return type> method name ( [arg list] )
{
 //Boc
 }
```

Main Points:

1) **Synchronization allows user to control access of Multiple Threads to any Shared Resource**
2) **It is used when you want to allow only one thread to access the shared resource.**
3) **Synchronization makes a lock on resource and release after completion of task.**
4) **It uses synchronized method using synchronized keyword.**
5) **It used to lock object for any shared resource.It automatically aquires lock for that object & release it when thread complete task.**
   Example: Synchronized Resource



Part-2 : File Handling ,Java.io package

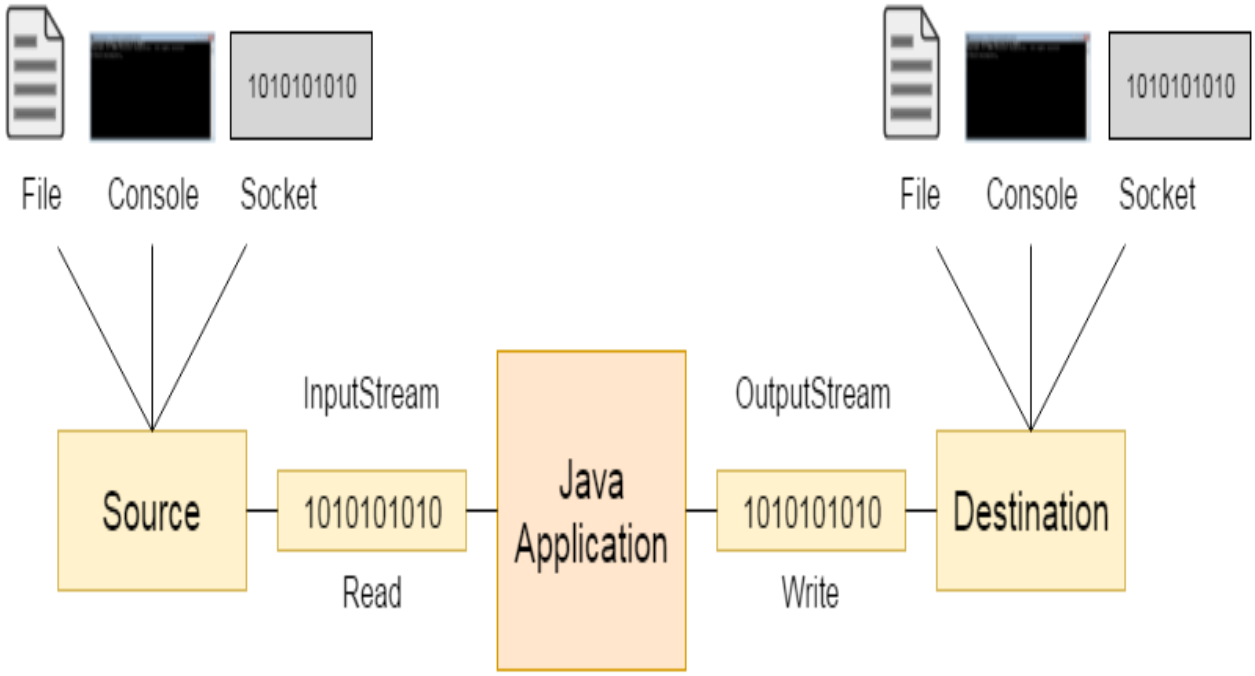
Streams Input & Output.

Q-1) **What is Stream in Java ? Explain Types of Stream in Java.**
ANS:
Java I/O (Input and Output) is used *to process the input* and *produce the output*.
Java uses the concept of stream to make I/O operation fast. The java.io package contains all the classes required for input and output operations.
We can perform file handling in java by Java I/O API.
The java.io package contains every class to perform input and output (I/O) in Java. All these streams represent an input source and an output destination. The stream in the java.io package supports many data such as primitives, object ,characters, etc.
Stream
A stream can be defined as a sequence of data. There are two kinds of Streams –
InputStream and OutputStream are the basic stream classes in Java.
InPutStream − The InputStream is used to read data from a source.

- OutPutStream − The OutputStream is used for writing data to a destination.

Why we need Stream Classes?
To perform read and write operation on binary files we need a mechanism to read that binary data on file/to write binary data (i.e. in the form of byte). These classes are capable to read and write one byte on binary files. That's why we use Stream Classes.

How Works InputStream&OutputStrem ?
Java application uses an input stream to read data from a source, it may be a file, an array, peripheral device or socket.
Let's understand working of Java OutputStream and InputStream by the figure given below.



Types of Stream.

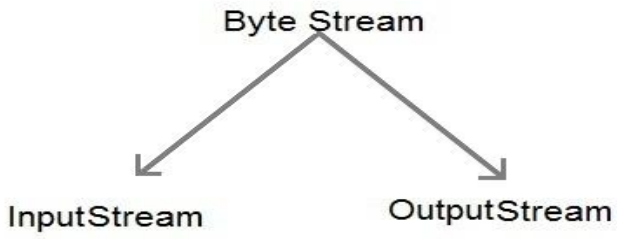
Java performs I/O through Streams. A Stream is linked to a physical layer by java I/O system to make input and output operation in java. In general, a stream means continuous flow of data. Java encapsulates Stream under java.io package. Java defines two types of streams. T

1. **Byte Stream** : It provides a convenient means for handling input and output of byte.
2. **Character Stream** : It provides a convenient means for handling input and output of characters. Character stream uses Unicode and therefore can be internationalized.

1)Byte Stream:

Java byte streams are used to perform input and output of 8-bit bytes. Though there are many classes related to byte streams but the most frequently used classes are, FileInputStream and FileOutputStream.

Byte stream is defined by using two abstract class at the top of hierarchy, they are InputStream and OutputStream.



Some important Byte stream classes.

Stream class	Description
BufferedInputStream	Used for Buffered Input Stream.
BufferedOutputStream	Used for Buffered Output Stream.
DataInputStream	Contains method for reading java standard datatype
DataOutputStream	An output stream that contain method for writing java standard data type

FileInputStream	Input stream that reads from a file
FileOutputStream	Output stream that write to a file.
InputStream	Abstract class that describe stream input.
OutputStream	Abstract class that describe stream output.
PrintStream	Output Stream that contain print() and println() method

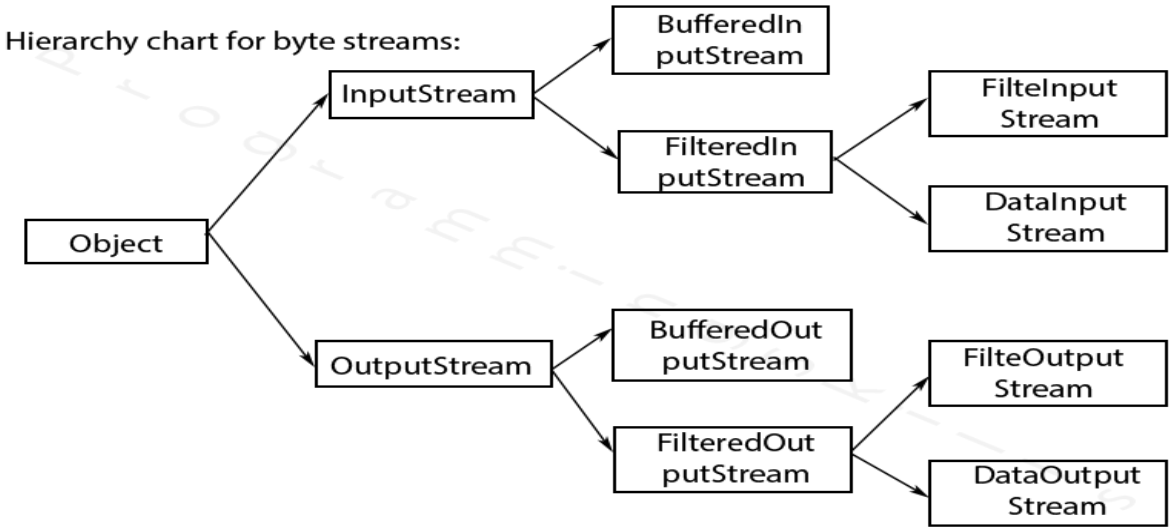
2) Character Stream:
java Byte streams are used to perform input and output of 8-bit bytes, whereas Java Character streams are used to perform input and output for 16-bit unicode. Though there are many classes related to character streams but the most frequently used classes are, FileReader and FileWriter
Character stream is also defined by using two abstract class at the top of hierarchy, they are Reader and Writer.

Some important Charcter stream classes.

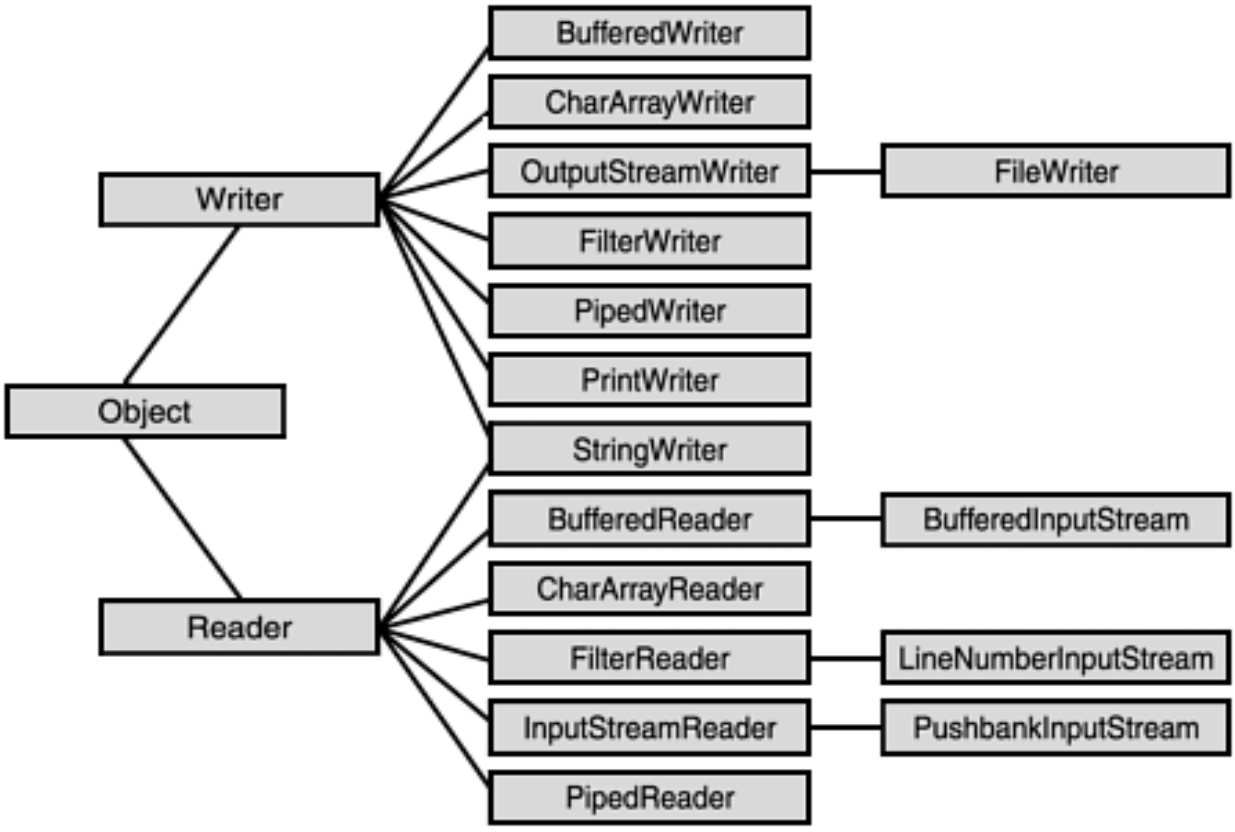
Stream class	Description
BufferedReader	Handles buffered input stream.
BufferedWriter	Handles buffered output stream.
FileReader	Input stream that reads from file.
FileWriter	Output stream that writes to file.
InputStreamReader	Input stream that translate byte to character
OutputStreamReader	Output stream that translate character to byte.
PrintWriter	Output Stream that contain print() and println() method.
Reader	Abstract class that define character stream input
Writer	Abstract class that define character stream output

Hierarchy of Stream

1) Byte Stream



2)**Character Stream:**



Q-2) File Class & its methods.

The File class of java.io package, is an abstract representation of file and directory structures. It is used to handle file and directory operations,

Class constructors

S.N.	Constructor & Description
1	File(File parent, String child) This method creates a new File instance from a parent abstract pathname and a

	child pathname string.
2	File(String pathname) This method creates a new File instance by converting the given pathname string into an abstract pathname.

- A file system may implement restrictions to certain operations on the actual file-system object, such as reading, writing, and executing. These restrictions are collectively known as access permissions.
- Instances of the File class are immutable; that is, once created, the abstract pathname represented by a File object will never change.

Field

Following are the fields for Java.io.File class −

- static String separatorChar − This is the system-dependent path-separator character, represented as a string for convenience.
- static char pathSeparatorChar − This is the system-dependent path-separator character.

File Methods

Modifier and Type	Method	Description
boolean	createNewFile()	It atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist.
boolean	canWrite()	It tests whether the application can modify the file denoted by this abstract pathname.String[]
boolean	canExecute()	It tests whether the application can execute the file denoted by this abstract pathname.
boolean	canRead()	It tests whether the application can read the file denoted by this abstract pathname.
boolean	isDirectory()	It tests whether the file denoted by this abstract pathname is a directory.
boolean	isFile()	It tests whether the file denoted by this abstract pathname is a normal file.
String	getName()	It returns the name of the file or directory denoted by this abstract pathname.
String	getParent()	It returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory.

Q-3) Random Access File Class

The Java.io.RandomAccessFile class file behaves like a large array of bytes stored in the file system.Instances of this class support both reading and writing to a random access file.

Constructors:

RandomAccessFile(File file, String mode)

This creates a random access file stream to read from, and optionally to write to, the file specified by the File argument.

Methods:

1	void close() This method Closes this random access file stream and releases any system resources associated with the stream.
	long length() This method returns the length of this file.
	long getFilePointer() This method returns the current offset in this file.
	int read() This method reads a byte of data from this file.
	char readChar() This method reads a character from this file.
	float readFloat() This method reads a float from this file.
	intreadInt() This method reads a signed 32-bit integer from this file.
	String readLine() This method reads the next line of text from this file.
	void seek(long pos) This method sets the file-pointer offset, measured from the beginning of this file, at which the next read or write occurs.
	intskipBytes(int n) This method attempts to skip over n bytes of input discarding the skipped bytes.
	void writeChars(String s) This method writes a string to the file as a sequence of characters.

Q-4) StreamTokenizer Class

The Java.io.StreamTokenizer class used to take an input stream and divide it into "tokens". The tokens to be read one at a time.

The stream tokenizer can recognize identifiers, numbers, quoted strings, and various comment styles.

Field

Following are the fields for Java.io.StreamTokenizer class −

- doublenval − If the current token is a number, this field contains the value of that number.
- String sval − If the current token is a word token, this field contains a string giving the characters of the word token.
- staticint TT_EOF − A constant indicating that the end of the stream has been read.
- staticint TT_EOL − A constant indicating that the end of the line has been read.
- staticint TT_NUMBER − A constant indicating that a number token has been read.
- staticint TT_WORD − A constant indicating that a word token has been read.
- intttype − After a call to the nextToken method, this field contains the type of the token just read.

Constructor:

StreamTokenizer(Reader r)
This creates a tokenizer that parses the given character stream.

Methods:

1)intlineno() : This method returns the current line number.

2)**intnextToken():** This method parses the next token from the input stream of this tokenizer.

Q-5) Piped Streams.

A pipe connects an input stream and an output stream.

A piped I/O is based on the producer-consumer pattern, where the producer produces data and the consumer consumes the data.

In a piped I/O, we create two streams representing two ends of the pipe. A PipedOutputStream object represents one end and a PipedInputStream object represents the other end. We connect the two ends using the connect() method on the either object.

We can also connect them by passing one object to the constructor when we create another object.

There are 2 ways to connect two ends of Pipe.

```
1)PipedInputStreampis  =newPipedInputStream();
PipedOutputStreampos  =newPipedOutputStream();
pis.connect(pos); /* Connect  the   two  ends  */

2)PipedInputStreampis  =newPipedInputStream();
PipedOutputStreampos  =newPipedOutputStream(pis);
```

1)PipedOutputStream: The PipedOutputStream class makes it possible to write to a Java pipe as a stream of bytes. Pipes are communication between threads running in the same JVM.

PipedOutputStream Example

```
OutputStream output = new PipedOutputStream(pipedInputStream);

while(moreData) {
int data = getMoreData();
output.write(data);
}
output.close();
```

The write() method of a PipedOutputStream takes an int which contains the byte value of the byte to write

2)PipedInputStream:

The PipedInputStream class makes it possible to read the contents of a pipe as a stream of bytes. Pipes are communication channels between threads inside the same JVM. Pipes are explained in more detail in my tutorial about Java IO Pipes.

PipedInputStream Example

```
InputStream input = new PipedInputStream(pipedOutputStream);

int data = input.read();
while(data != -1) {
  //do something with data...
doSomethingWithData(data);

data = input.read();
}
input.close();
```

The read() method of a PipedInputStream returns an int which contains the byte value of the byte read.
If the read() method returns -1, there is no more data to read in the stream, and it can be closed. That is, -1 as int value, not -1 as byte value. There is a difference here!

.

Q-6) Bridge Class:InputStreamReader & OutputStreamWriter Class in java.

1)InputStreamReader :

The Java.io.InputStreamReader class is a bridge from byte streams to character streams.It reads bytes and decodes them into characters using a specified charset.

Constructors:

InputStreamReader(InputStream in)
This creates an InputStreamReader that uses the default charset

Methods:

1) void close()

This method closes the stream and releases any system resources associated with it.

2) int read()

This method reads a single character.

3) int read(char[] cbuf, int offset, int length)

This method reads characters into a portion of an array.

2)OutputStreamWriter :

The Java.io.OutputStreamWriter class is a bridge from character streams to byte streams. Characters written to it are encoded into bytes using a specified charset.

Constructor:

OutputStreamWriter(OutputStream out)
This creates an OutputStreamWriter that uses the default character encoding.

Methods:

1) void close()
   This method closes the stream, flushing it first.
2) void flush()
   This method flushes the stream.
3) void write(int c)
   This method writes a single character
4) void write(String str, int off, intlen)
   This method writes a portion of a string.

Ch-4 &5 GUI using Applet &JComponents(Swing)&LayoutManager

Q-1) Write a note on Applet. OR What is applet?

- ✓ Applet provides graphical user interface mostly it reterenced in a webpage.
- ✓ An applet is used to build simple program which contains string , shapes & also images.
- ✓ So, applet is a program which can referenced by HTML source of a webpage.
- ✓ To execute applet , you need to a web browser of a tool which is called "appletviewer".
- ✓ Appletviewer is included in JDK. The superclass for applet is java.applet.Applet;
- ✓ Applet is defined by java comment
  Example:- /*<applet code="" width="0" height="0"></applet>*/
- ✓ This command determines the HTML source code.
- ✓ This tag determines which class inherit in the applet.
- ✓ It also define width & height of applet to display.
- ✓ Appletviewer executes an applet using HTML source code.
- ✓ Applet also can be accessed by web browser so nowdays. Applet program is used in net.

Exa:-
```
import java.applet.*;
    import java .awt.*;
/*<applet code="tmp" width="300" height="300"></applet>*/
public class tmp extends Applet
{
    String s="welcome";
    Public void paint(Graphics g)

    Public void paint(Graphics g)
    {
        g.drawString(s,100,100);

     }
}
```

To Compile >javac tmp.java
To run >appletviewer tmp.java

Q-2) Define life cycle of applet.

- ✓ Every java application begins execution within main().but applet by also browser or tool like appletviewer. So , it doesn't contains main().
- ✓ Applet remains in like state until in begins to close in between it trasforms in four state.
- ✓ This four state of applet is called "life cycle of applet".

I. **init()**
II. **start()**
III. **stop()**
IV. **destroy()**

1.init():-

- ✓ This method is called only once when applet begins the execution.
- ✓ This method contains information which can be executed for only once such as title for file etc…

2.start()

- ✓ This method executed after init method.
- ✓ It is called by appletviewer to resume the applet.

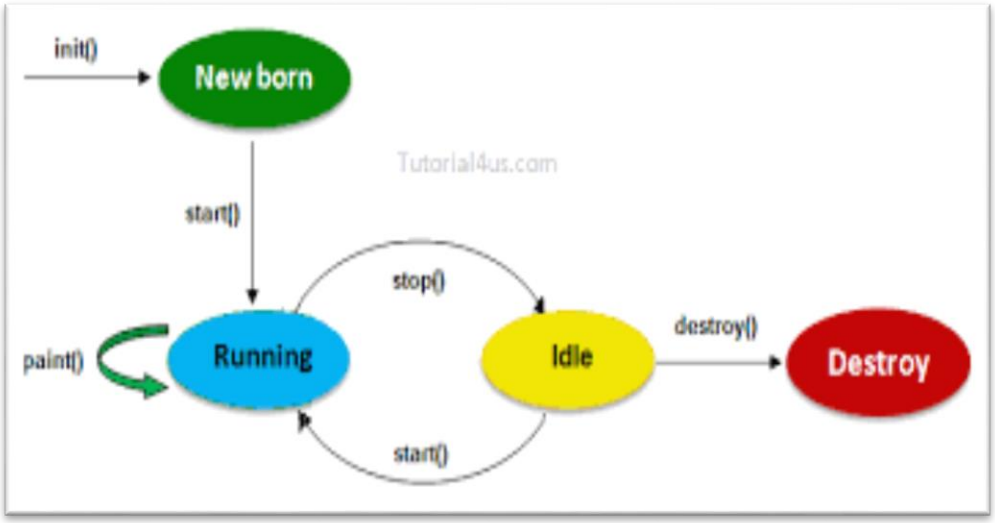
- ✓ Start() is called multiple times during it's life cycle line.
  a. If you go to another page from current page.
  b. If you maximize the minimized applet.

3.stop()

- ✓ This method is called when applet is suspended for a while.
- ✓ It is also called when applet is long minimized.
- ✓ So, stop() is also called for multiple times.

4.destroy():-

- ✓ This method is automatically invoke when applet is closed or terminated.
- ✓ It is called after stop().
- ✓ So it is only for once during it's life cycle.



-3) What is job of awt package? And it's classes.

- ✓ AWT stands for Abstract Window Toolkit.
- ✓ This package provides all graphical & formatting tools which makes applet interative like the it provides different shape ,color& also fonts.
- ✓ Awtpacakage contains different types of class which are as follow.

A) Define Font class

- Font class has ability to control the appereance of string.
- It defines different types of font styles.
- Font class determines the size of font and of characters.
- The class is java.awt.font

Constructor:-

- Font(string font,intstyle,int size);
- String font determines the fontname like arial,arialblack,times new roman,etc..
- Style indicates the bold ,italic,normal and bold italic.
- Size determines the font size in points.

Methods:-

1. setFont():-
   - It contains the font object to provides font effect .
   - When it is set after that string will be affected.
   - Syntax:void setFont(font f);
2. stringWidth():-
   - It return the width of the specified string.

- Syntax:int stringWidth(string s);

3. getAscent():-
- this method I used to return the ascent value.
- Syntax:int getAscent();

4. getDescent():-
- this method is used to return descent value.
- Syntax:int getDescent();
- When string is displayed,allcharactes are assighne into horizontal line.
- You can align character to above or below line using ascent&descent.
- Java provides font library class to get the diffrence size & font.

5. matrics():-
- syntax:Font matrics(font f);
- ex: Font f=new Font("Arial black" ,1,30);
  g.setFont(f);

B) Define Dimension class

- demention class determines the size & applet in width & height.
- So,this class provides dynamic determination of dimention.

Constructer:

- Dimension(Dimension d):
- it represents total width & height in digit.
- Dimension(int w,int h):
- It determines width & height indvidualy.

Methods

1. Dimension getSize():-
   - It is used to return the size of an applet.

C) Define Color class

- This class provides different colors .
- This color can be used by each object with particular color.
- You can set the color for any shape,background& also foreground.
  - Color(intr,intg,int b):-It defines arrange from 0 to 254.

I. Color(intrgb):-

II. Color(float r,floatg,float b):-
  - It takes float value between 0.0 to 1.05.

Methods

1. setColor(color c):-
   - It is used to set the color for instance.
2. intGetBlue():-
   - It returns blue component of current object.
3. intGetRed():-
   - It returns red component of current object.
4. IntGetGreen():-
   - It returns green component of current object.
5. IntGetRGB():-
   - It returns the mixture of red,green,blue.
   - The package of color is java.awt.color.

4) Define Graphics class

1.Graphics:-

- ✓ All during tools like string & shapes are provide by graphics class.

- ✓ This facility is imported by import java.awt.Graphics.
- ✓ Graphics class provides a set of method which caperform graphics ouput.

Methods:

1. Void drawRect(intx,int ,intw,int h):-
   - It draws the rectangle at x & y position for specified width & height.
2. Void drawOval(intx,inty,intw,int h):-
   - It draws the oval of the x & y for specified height or width.
3. Void drawLine(int x1,int y1,int x2,int y2):-
   - It draw the line if (x1,y1) to (x2,y2) position.
4. Void drawArc(intx,inty,intw,inth,int d1,int d2):-
   - It draws the arc full (x,y) with specified w & h declare degree 1 to degree 2.
   - A degree is set to 3 or 2 watch & so on.
5. Void drawPolygon(int x[],int y[],int points):-
   - This method is used to define polygon shapes with number of points.
6. Color getColor():-
   - This method returns the color.

7 .void drawString(String,intx,int y)

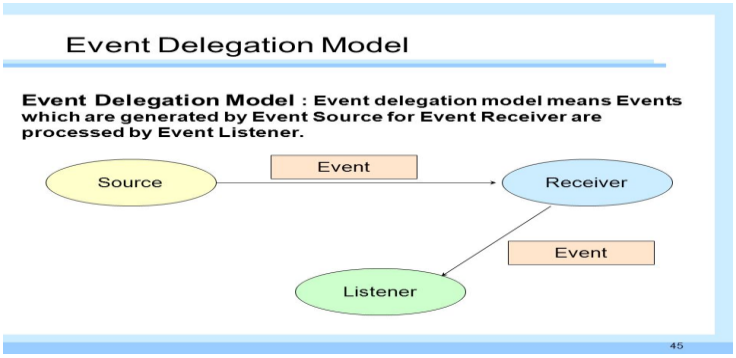
- It display the string at given location of x & y which includes horizontally & vertically.

8 void drawImage(Image i1,int x,int y, int w,int h,CurrentReference)

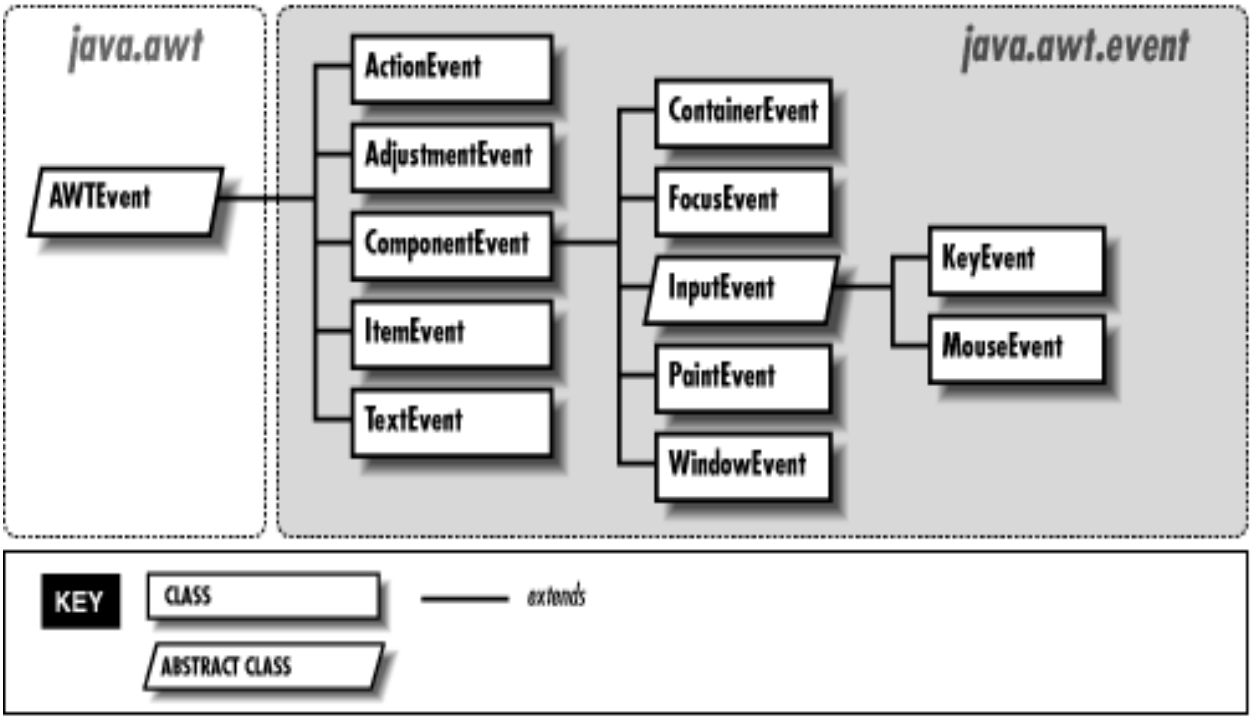
- It is used to drawImage over the Applet.It takes Image Resource, x & y coordination, total width & height of an Image and currentReference object of same class.

Q-4) Event Delegation model.

- ✓ Awt provides a set of standard componets which is used to interect between user & java applet.
- ✓ This component [control] provides different action which is known as event.
- ✓ Applet is an event driven .so user can built GUI based event driven java application.
- ✓ Java provides many event interface supported by event model.
- ✓ The collection of event model is known as delegation event model.
- ✓ The delegation model is capable to generate event from source to its listener.
- ✓ Any event has two responsibilities.
  a.sourcegenerater
  b.listenergenerater.
- ✓ So , event is an object which is executed when state of source is changed.
  1. Source generater:-
- ✓ There are three responsibilities of source generater.
  1.it must provide a method which allows listener to register event.
  2.it must have register specific type of listener which generate event
  3.it must sent an event to it's specific listener.
  2. Listener generater:-
- ✓ Listener generator must be register to received the event.
- ✓ It must registed interface to receive different types of event.



Hierarchy Of Events:



Q-5) Explain different types of event in awt.

1. **ActionEvent:-**
   - ✓ When user press any button that time action event is generated.
   - ✓ It also fired when enter key is pressed in textbox.
   - ✓ ActionEvent must be registered in "ActionListener" because it is derived from action listener.
     Synatax**:[component/object].addTypeListener[object]**
     Example: b1.addActionListener(this);
   - ✓ This listener must provides a method which is must be override.

Syntax:public void actionPerfomed(ActionEvent e)
{
//boc
}

2. **MouseEvent:-[MouseListener]**
   - ✓ Whenever your mouse is dragged , moved , prssed , released and also mouse entered and mouse exit from specific area that time **MouseEvent** is generated.
   - ✓ It provides **MouseListener** to receive mouse events
   - ✓ It provide many overridden methods.
     1. Void mouseClicked (MouseEvent)
     2. Void mouseEntered (MouseEvent)
     3. Void mousePressed(MouseEvent)
     4. Void mouseReleased(MouseEvent)
     5. Void mouseExited (MouseEvent)

     - ✓ Mouse provides many special method to returns the co-ordinate of mouse.
       1. getx():- it returns x co-ordinate.
       2. gety():- it returns y co-ordinate.
       3. clickCount():- it return total number of clicked event is fired.
       4. MouseAdapter class:- it provides MouseAdapter class for mouse event.
3. **TextEvent:-**
   - ✓ When event the value of text and text area is changed that time text event is fired.
   - ✓ It provides **TextListener.**
   - ✓ It has only one method to receive TextListener.
   - ✓ Syntax:- void textValueChanged(TextEvent e)
4. **FocusEvent:-**
   - ✓ Whenever any controls get the focus & the last focus that time focus event is generated.
   - ✓ It has "**FocusListener".**

   Method:-
   1. void focusGained (FocusEvent)
   2. void focusLost (FocusEvent)
5. **AdjustmentEvent:-**
   - ✓ Whenever serollbar is manipulated that time this event is generated.
   - ✓ It has "**AdjusmentListene**r".

   Method:-
   public void adjustmentValueChanged (AdjustmentEvent)
6. **ItemEvent:-**
   - ✓ Whenever the item of listbox, checkbox, option button & also choice is clicked or selected that time item event is generated.
   - ✓ It has "**ItemListener**".

   Method:-
   void itemStateChanged (ItemEvent)
7. **WindowEvent:-**
   - ✓ Whenever window is opened, activated, deactivated, close &iconified that time window event is generated.
   - ✓ It provides **"WindowListener"** & **WindowAdapter** class.

   Methos:-
   - ✓ Void windowOpened (WindowEvent)
   - ✓ Void windowClosed (WindowEvent)
   - ✓ Void windowClosing(WindowEvent)

- ✓ Void windowIconified (WindowEvent)
- ✓ Void windowDeiconified (WindowEvent)
- ✓ Void windoeActivated(WindowEvent)
- ✓ Void windowDeactivated (WindowEvent)

8. KeyEvent:-

Whenever keypressed ,keydown, keytyped, that time key event is fired.
It has "**keyListener" &keyAdapter** class.
Methods:

- ✓ **Void keyReleased(KeyEvent)**
- ✓ **Void keyPressed(KeyEvent)**
- ✓ **Void keyTyped(KeyEvent)**

9. ComponentEvent:-

- ✓ Whenever any component is moved, reside hide & visible that time component event is fired.
- ✓ It has **componentListener** & **ComponentAdapter.**
- ✓ It has getComponent();

10. ContainerEvent:-

- ✓ Whenever any component is add to the container & removed from the container that time this event is fired
- ✓ **It has 2 methods:-**

1. add (component)
2. remove (component)

Q-6) Explain awt controls & Swing Controls

1)AWT Controls

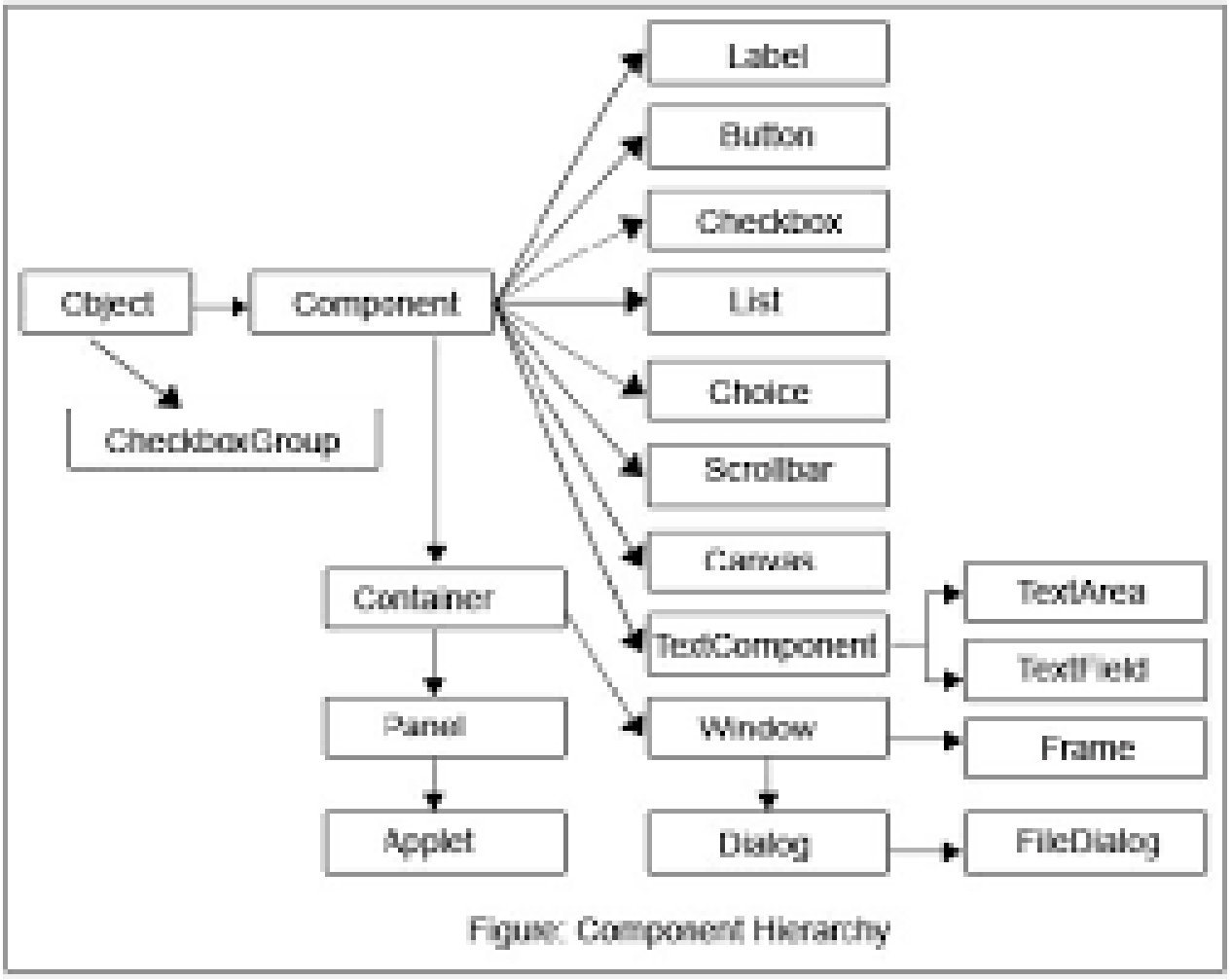
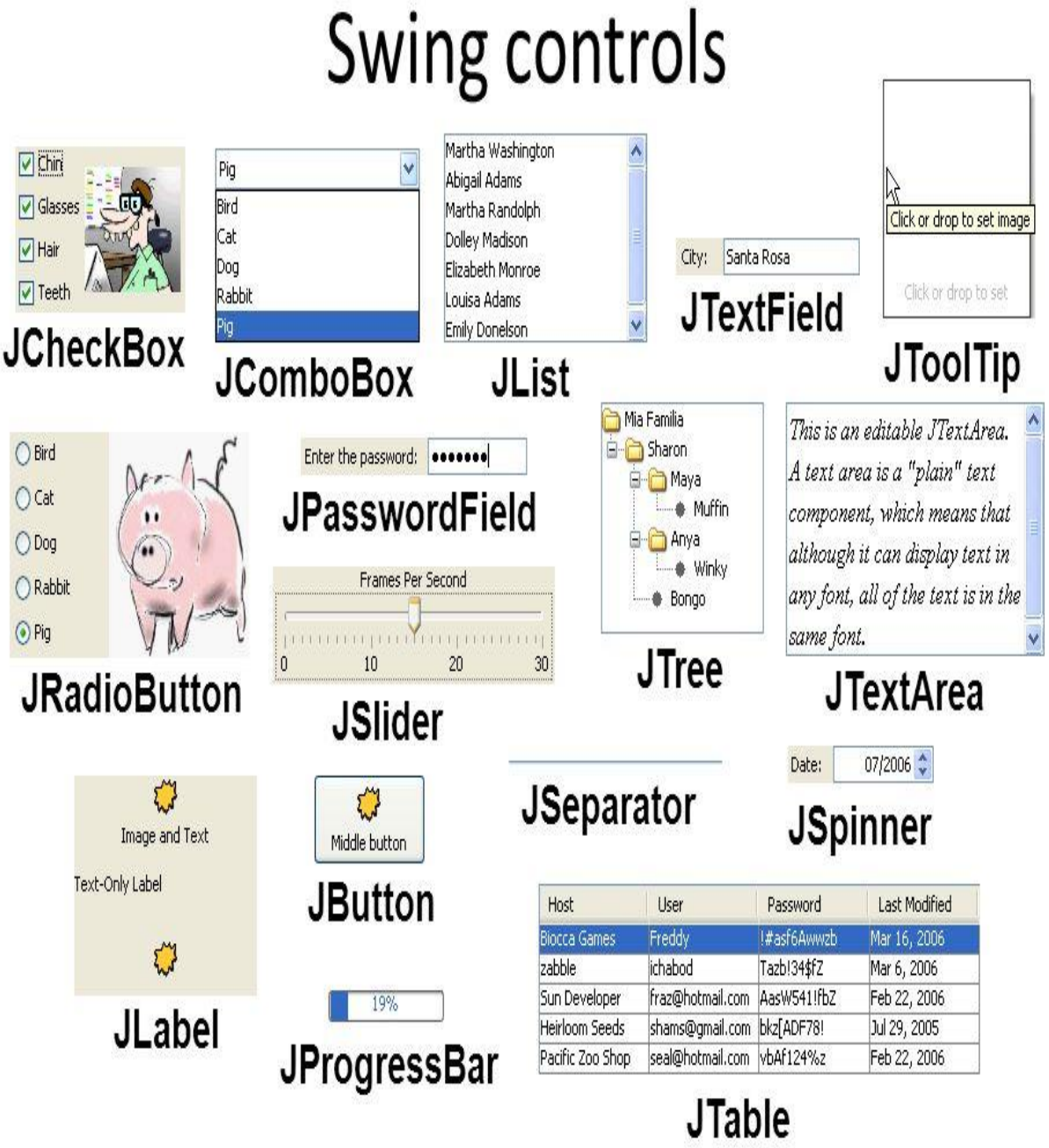


Figure: Component Hierarchy

2) Swing Controls



Note: Both AWT & Swing control

Explanations are Given

- ✓ Awt provides many controls such as label, button, textbox, etc..
- ✓ This controls provides user interface to built customize application
- ✓ Swing also provide controls of awt& also advance tools like table, grade, etc..

1) Label controls:-

- ✓ Label provides a facility to display the information to the user.
- ✓ It does not allow to take input from the user.
- ✓ Label can not support any user enteraction. Inshort it is not capable to first event.

- ✓ It has 3 constructor:-
  1. Label()
  2. Label(string info)
  3. Label(string element)

JLabel control:-

- ✓ It is a component of JAPPLET.
- ✓ It is provide by swing class.
- ✓ It allows user to display image icon into the label as a background.
- ✓ It has 3 constructor:
  1. JLabel (icon)
  2. JLabel (string)
  3. JLablel (string,icon,element)

Methos:-

1. **getText():-**
   - ✓ this method return the text of the label.
   - ✓ Syntax: string getText();
2. **setText():-**
   - ✓ this method is used to set text on the label.
   - ✓ Syntax:- void setText(string);
3. **getIcon():-**
   - ✓ it returns the icon file set in lable.
4. **setIcon():-**
   - ✓ it is used to set the icon file to a lable.

2).TextField:-

- ✓ TextField allows user to take input from user.
- ✓ TextField only obtain single line.
- ✓ It is capable to generate “ActionEvent”,”KeyEvent” & “FocusEvent”.
- ✓ It has 4 constructor:
  1. TextField()
  2. TextField(string)
  3. TextField(int size)
  4. TextField(string , int size)

J Textfield:-

- ✓ It received from component.
- ✓ It provides some facility of TextField.
- ✓ Constructor are same as TextField.

Methods:-

1. **getText():-**
   - ✓ it returns the text from the text box.
   - ✓ Syntax: String getText();
2. **setText():-**
   - ✓ it set the text into text box.
   - ✓ Syntax:void setText(String);
3. **selectAll():-**
   - ✓ it select all the contents of the TextField.
4. **getSelectedText():-**
   - ✓ it return the selected text from the TextField.
5. **select(int start, int end):-**
   - ✓ It select the text from specified position start to end.
6. **setEditable():-**

- ✓ it set the text n textbox in edit table or non edit table mode.
- ✓ By default it is true.

3)TextArea():-

- ✓ TextArea is a multiline area textbox which provides textbox.
- ✓ It contains a text area if greater size of text area , then it autometicalliy add scroll bar.
- ✓ It has 4 constructor:
  1. TextArea()
  2. TextArea(string)
  3. TextArea(introw,int col)
  4. TextArea(string, int row, int col, scrollbar)
     - ✓ It support all the method of textfiled but it has 3 special method.

Methods:-

1. **append():-**
   - ✓ It append the string to the end of the text area.
2. **insert(String, int start):-**
   - ✓ It insert the stirng to the specified position.
3. **replaceRange(new string, int start, int end):-**
   - ✓ it replace the string at given position from start to end.

4) Button control:-

- Button provide a facility to generate ActionEvent.
- It is also known as PUSH BUTTON.
- it has two constructor:

1)Button()
2)Button(String)

***J BUTTON**:-

- it also provide a same functionality of a button but it allows user to embeded string & icon on the button.
- It has three constructor:-
  1)JButton (Icon)
  2)JButton (String)
  3)JButton (String,Icon)

=>Methods:-

1)String getLabel():-

- It return the label or String of the button.

2)void setLabel(String):-

- It also used to set the text on the button.

5) Checkbox control:-

- Checkbox is used to select multiple item from the list of item.
- Checkbox is a combination of label & box.
- It has two state:
  1)checked
  2)unchecked
- it has four constructer:
  1)Checkbox()
  2)Checkbox(String)
  3)Checkbox(String,boolean state)

4)Checkbox(String,booleanstate,checkbox group)

***JCheckbox:-**

- The JCheckbox class provide advance facility of checkbox.it is also known as toogle button.
- It has six constructor:-

1)JCheckbox(icon)
2) JCheckbox(String)
3) JCheckbox(String,boolean state)
4) JCheckbox(icon,boolean state)
5) JCheckbox(String,icon)
6) JCheckbox(icon,String,boolean state)

=>Methods:-

1) getLabel():-
- This method return the String of the checkbox.

2)getState():-
- This method return the status of the checkbox either true or false.

3)setLabel():-
- This method is used to set the label for checkbox.

4)setState():-
- It is used to set state of checkbox.

6)Checkbox Group:-
- Checkbox group contain the collection of checkboxes. It provides a facility to select only one checkbox from group of checkbox.
- At a time only one checkbox is selected that time no other checkbox is selected.
- It is also called "option button".

=>Methods:-

1)getSelectedCheckbox():-
- It return the selected checkbox from the given list.

2)setSelectedCheckbox(checkbox):-
- This method is used to set selected checkbox from the list of checkbox.

7) JRadio Button:-
- It also provides a facility to select only one item from the list.
- It has two state.
  1)true
  2)false
- JRadio button is embeded into JApplet.
- It has following constructor:-
  1)JRadioButton(Icon)
  2) JRadioButton(String)
  3) JRadioButton(Icon,boolean)
  4) JRadioButton(String,boolean)
  5) JRadioButton(String,Icon)
  6) JRadioButton(Icon,String,boolean)

NOTE:-Japplet provides ButtonGruop class to collect or initialize JRadioButton.
SYNTAX:-ButtonGroup b=new ButtonGroup()
b.add(RadioButton)

8)List Control:-
- A List control allows user to create list of more than one item.
- If item are larger than Listbox then java automatically add scrollbar fro your entire list.
- List control provides two events.

1) ActionEvent:-
   - Whenever you double click on the item it action eventlist is get fire.
2) ItemEvent:-
   - When you select item or deselect item that time item event is generated for list.
   - It has three constructor:-
     1)List()
     2)List(int row)
     3)List(int row,boolean multiple)

- Boolean multiple:-
  Multiple arguement determine to select more than one time if it is set to true.

=>Methods:-

1)add(String):-
- It add the item to the end of list.

2)add(index):-
- It add the item to the specific index.

3)getItem():-
- It return the name of item in specified index.

4)getItemCount():-
- It return the total no of item from the list.

5)getSelectedIndex():-
- It return the index of selected item.

6)getselectedIndexes():-
- It return the index of multiple selected item.

7) isMultipleMode():-
- It return true if multiple item will be selected otherwise return false.

8)remove(String):-
- It remove the item of specified index.

9)remove(int index):-
- It remove the item of specified index.

10)removeAll():-
- It remove entire list.

JList:-

- **T**his control allows user to create list of items,it also provides a facilty to select any one or more item.
- This control does not have Scrollbar.so, this control is first loaded into J-scroll pane control.
- This control can't be modify during runtime.

***constructor:-**

i)JList()
ii)JList(vector[])
iii)JList(object[])

- Its method same as a List.
- Though inner class mechanism , user can direct access any method & any variable in outer class.
- Inner class also extends Adapter class.
- This class also provides a facility to use any methods of listener you don't need to implement all methods.

9) Choice Control :-

- Choice control also provides creation of list of item.
- But it provide dropdown listbox in which you can select only one item.
- There is one advantage of choice control.it requires less space into contains but store many values.

***Cons**tructor:-

1)choice():-

=>Methods:-

1)addItem(String):-

- It add the given item into choice control.

2)String getItem(int index):-

- It return the string of specified index.

3)getSelectedIndex():-

- It return the index of selected item.

4)getSelectedItem():-

- It return the selected item.

5)insert(String,index):-

- Insert the String into given position.

6)remove(Index):-

- it remove the String from given index.

7)removeAll():-

- it is used to remove all the String.

8)select(index):-

- it return item from specified index.

9) select(String):-

- it return item from specified string.

=>JComboBox:-

- JComboBox control is define by swing class.
- It provides drop-down list box like choice controls.
- It has two contructors:-
  1)JComboBox()
  2)JComboBox(vector)
  - Vector determine the set of value inserted into combobox.
  - It provide same method as choice control.

10)Scrollbar:-

- Scrollbar component provide a facility to select item as an integer value between in minimum range & maximum range.
- Scrollbar contain shifts whenever it is drag that time it value is change & it generate adjustment event.
- It has three constructor:-
  1)Scrollbar()
  2)Scrollbar(int orientation)
  3)Scrollbar(int orientation,value,width,minValue,maxValue)
  - Orientation determine weather a scrollbar it is displayed horizontal or vertical.

=>**Methods:-**

i):-getValue():-

- It return the value of the scrollbar.

ii)setValue(integer):-

- It set the value for scrollbar with specified value.

Q-7) WRITE A NOTE ON MenuBar ,Menu & MenuItem:-

***MenuBar:-**

- MenuBar contains the place for different Menu.
- A MenuBar is a location which display a list of top level menu choices.
- Each window is associated with MenuBar.
- To create a MenuBar there is a MenuBar class.

=> class MenuBar():-

- This is used to create MenuBar of given menu.
- This MenuBar will be registered into the applet by action.
- Syntax:- setMenuBar(Name)

***Menu:-**

- Menu is displayed on the menubar.
- One MenuBar contains many Menus.
- Each menu has drop-down list.

o Menu is creating with following constructor.

->MenuItem(String name):-

- MenuItem is a list of submenu which is displayed in top menu.

=>Constructor:-

o MenuItem(String name)
o Each MenuItem must be added in Menu.

=>Method:-

menuName.add(MenuItem nm)

- File Edit
  Open cut
  Save copy
  Close paste

Q-8)Write a note on Layout manager:-

There are different types of layout.

1)FlowLayout
2)BorderLayout
3)GridLayout
4)CardLayout
5)GridBag Layout
6)Panel

Layout manager provides significant benefit.

i) it is not necessary to calculate the co-ordinate at which an element should be placed.
ii) A layout manager dynamically adjust the placement of the elements if an Applet is resize.
iii) component have different sizes on diffrent platform therefore it is not possible to specify one set of co-ordinate. That will be suitable in all environment the layout manager handle this class.

A layout manager is associated with each container object the methods to get & set are as follow:-

- **LayoutManager getLayout()**
- **Void setLayoutManager(LayoutManager)**

1)FlowLayout:-

- It is the default layout manager components are layout from the upper left corner , left to right & top to bottom.
- Constructor:-
  1)FlowLayout()
  2)FlowLayout(int align)
  3)FlowLayout(intalign,inth.space , int v-space)

2)BorderLayout:-

- It allows a program to specify the placement of component. This is done by using the graphics terms that is “north”,”south”,”west” “corner”.
- Constructor:-
  1)BorderLayout()
  2)BorderLayout(int h-space , int v-space)

3)GridLayout:-

- The grid Layout class automatically arranges components in a grid all components are created with equal size.
- Constructor:-
  1)GridLayout()
  2)GridLayout(int row, int col)
  3)GridLayout(int row, int col, int h-space , int v-space)

=>Insets:-

- The insets class encapsulates information about the top,left,bottom or right margin around the boundary or a container.
- **getInsets (int top, int left , int bottom , int right)**
- **Method:- getInsets()**

4)CardLayout:-

Constructor:

i)CardLayout()
ii)CardLayout(int h-space , int v-space)

- The cards are typically held in object of type panel.

i)create a panel that contains a clock & a panel for each card with deck.
ii)An appropriate panel that components for each card.
iii)Add this panels to the panel for which card layout is the layout manager.
iv)Add this panel to the main Applet.

- Void first(Container deck)
- Void last(Container deck)
- Void next(Container deck)
- Void previous(Container deck)

5)GridBagLayout:-

- In GridBag layout you can specify the return placement of component by specifying their position within call inside the grid.
- The key to the gridbag is that each component can be a different size & each row in the grid can have a different number of columns.
- Constructor:-
  i)GridBagLayout()
  ii)GridBagLayout(2,3)
    - It is not possible in grid but possible in GridBagLayout.

Q-9) WHAT IS JAPPLET:-

- JApplet class is extends Applet class.
- JApplet provides graphical area in which swing component can be placed.
- JApplet provides extra functionality like it supports panels , glass panels & root panels.
- JApplet also uses add() to add the component on to the JApplet.
- Syntax:- Void add(component)

Explain JTabbedPane:-

- A Tabbed pane is a collection of different folders & one file.
- Each folder is known as tabbed & each tabbed
  Has title.
- Whenever user selects a particular tabbed that time its related contains will be displayed.
- Only one tabbed is selected from the given group of tabbed at time.
- This type of mechanism is usually used in setting or configuration options.
- It uses addtab() to add tab in the group.
- Syntax:- void addTab(title, component)

JTable:-

- JTable component displays the data as a combination of row & column.
- It also allows uses to resize the column by dragging cursor of the column.
- This facility is provided by JTable class.
- Syntax:-
  JTable(obj a[] , str heading[])
- First argument contains the actual data to be displayed on the table.
- This data is stored in a dimentional array.
- Its datatype is object because there is a mixed datatype value in table.
- Second argument contains heading of the each column.

Q-10)What is swing? How different of AWT:-

- Swing is a also a component but it is not implemented for platform specific code like AWT.
- Swing is platform independent & it provides all the control of AWT & also advance control.
- Swing class use all the event of AWT & all the method of AWT controls.
- Swing class is placed in java x package.so, java extention package is call javax.
- Swing class provides more extra functionality then AWT components,some different are given below.

Swing	**awt**
swing button & label can display & string & also can display image as background.	Awt button & label can not display image in background.
Swing class provides extra appearance facility that means button can be display as around.	In awt all component must have in rectangular.
If provides some extra controls to perform customize calculation like JApplet,JTable,JTree etc.	Some Advance control is not available in awt.

Q-11) What is AnonymousInner class?

- It is also an inner class but it doesn't have a name of the class.
- It directly create the instance of adapter class by following syntax.
- Syntax:- new<classnm>()

```
{
    //boc
}
```

  - Class name is a name of Super class of adapter.
  - Which methods you want to use is defined between {}.
  - The scope of anonymous inner class remains for that control only. Which has define anonymous inner class.