Unit - 3.1

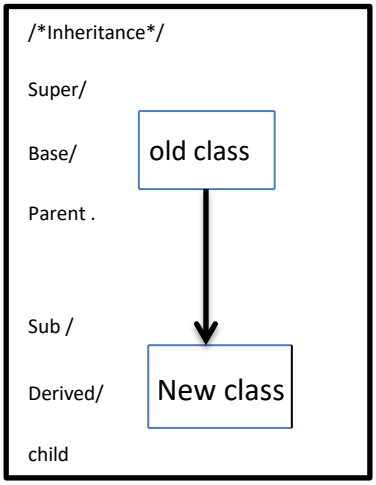
Inheritance

1. what is inheritance? Explain types of inheritance.

A. **Inheritance:** The mechanism of deriving a new class from old class is called inheritance.

The old class is called as super class/ Base class or parent class.

The new class is called as derived class/ sub-class/ child class.



The inheritance allows sub classes to inherit, the all variables and methods of their parent class.

The Inheritance concept is used to reuse the variables and methods of class.

The inheritance allows the reusability of variables and methods of the class.

Defining the subclass:

SYNTAX:- class subclass name extends superclass name

{

Variable declaration;

Method declaration;

}

→In the above syntax subclass name, superclass name are java valid identifiers.

→Extends is a keyword, and it is used to extend the properties of superclass to the sub class.

→The subclass will contain its own variables and methods as well as variables methods of superclass.

→In the inheritance we can add more properties to a Existing class without modifying class.

Types of inheritance:-

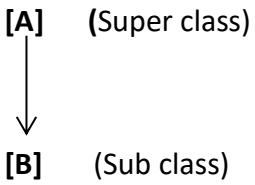
Inheritance is classified into 4 types.

1. Single inheritance.
2. Multiple-inheritance.
3. Multi-level inheritance
4. Hierarchical inheritance.

Single inheritance:-

The process of deriving the only one super(new) class from one base class is called single inheritance.

In the single Inheritance only one Super class and one sub Class.



Syntax:-

/*single inheritance*/

Class superclass name

{

Variable declaration;

Method declaration;

}

Class subclassname extends superclassname.

{

Variable declaration;

Method declaration;

}

}

Example:-

Class A

{

Variable declaration;

Method declaration;

}

Class B extends A

{

Variable declaration;

Method declaration;

}

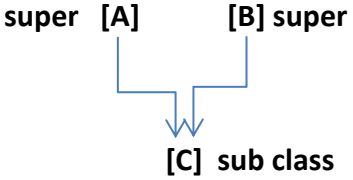
Description:-

In the above Syntax sub class name and superclass name of Java valid identifiers.

The sub class contains variables and methods of super class.

Multiple Inheritance:-

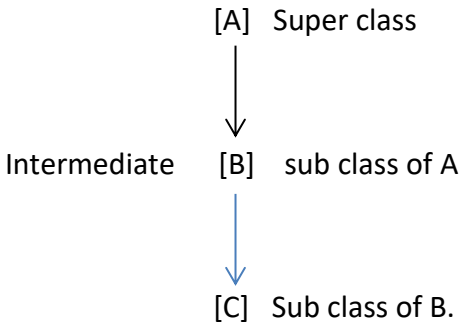
The process of deriving the new class from more than one old class is known as multiple inheritance.



Java does not support multiple-inheritance. But it can be implemented by using interface.

Multilevel -inheritance:-

The process of deriving the new classes from old class is called multilevel inheritance.



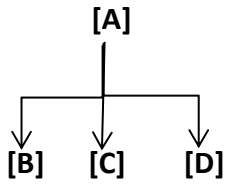
Syntax :-

```
Class A
{
Variable Declaration;
Method declaration;
}
Class B extends A
{
Variable declaration
Method declaration;
Class C extends B
{
Variable declaration;
Method declaration;
}
```

→In the above Syntax class A, B and C are valid Java identifiers. The sub class c contains its own variables and methods as well as variables and methods of both A and B.

Hierarchical inheritance:-

The process of deriving more than one a new class from one old class is called hierarchical inheritance.



The declaration of hierarchical inheritance as follows.

Syntax:-

/*Hierarchical Inheritance*/.

Class A

{

Variable declaration;

Method declaration;

}

Class B extends A

{

Variable declaration;

Method declaration;

}

Class C extends A

{

Variable declaration;

Method declaration;

}

Class D extends A

{

Variable declaration;

Method declaration;

}

Description:-

In the above Syntax class a, b, c and d are valid Java identifiers.

In the above declaration class b c and d are the subclass of superclass A.

Subclass constructor:-

1. The sub class constructor is used to initialize (construct) instance variables of both super and subclasses.

2. The class constructor uses keyword super constructor method of the super class.

3. The super keyword is used then follow the following rules:

I Supper May only be used with in a subclass constructor.

II. Call the superclass constructor must be appear as first statement with in subclass constructor.

III. The parameter list in the super call must match the order and type of instance variable decrease in the superclass.

2. Explain about method overriding in Java.

A. Method overriding:-

1. Redefine the superclass method within the subclass is called overriding.

2. Defining a method in the supplies they method having the same name same parameters list under same return data type as same method in super class.

3. When that the super method is called a method defined in the subclass is called and executed instead of superclass method this is called overriding.

Example :-

```
import java.io.*;
class A
{
int x;
A(int p)
{
x=p;
}
void display()
{
System.out.println("super"+x);
}
}
class B extends A
{
int y;
B(int p,int q)
{
super(p);
y=q;
}
void display()
{
```

```
System.out.println("superclass"+x);

System.out.println("superclass"+y);

}

}

class c

{

public static void main(String args[])

{

B b1=new B(10,20);

b1.display();

}

}
```

3.Explain about final variable, final method and final class.

A. **Final:-**

1. Final is a keyword.

2. Final is a modifier, It is the used to declare the variables methods and classes.

3. The variables methods and classes are declared by using final keyword is called the final variables final method and the final classes respectively.

Final variables:-

1. All variables and methods can be used to override by default in subclass.

2. Final variables are constant whose value does not be changed.

3. The members of the superclass and subclass can be declared using final ”keyword”.

4. The final variables are used outside of the class.

5. Final variables are replaced during compile time.

6. Final variable values are initialized by programmer.

7. No, Default values are assigned to final variables.

Syntax:-

Final datatype variable name = value;

Eg: final int x= 10;

Final method:-

1. Final methods are methods that are not-override by the methods of the subclass.

2. Final methods are not override.

Final class:-

1. A class that cannot be a subclass is called a final class.

2. The final class cannot be extended.

3. The new class does not derived from final class.

Syntax:-

Finalclass Classname

{

Variable declaration;

Method declaration;

}

4. Write about abstract methods and abstract class.

A. **Abstract:-**

1 Abstract is a keyword.

2. Abstract is a modifier that is used to declare the methods and classes.

Abstract method:-1. The method is declared without body using abstract keyword is called Abstract method.

2. The abstract method does not have body of the method.

3. We can indicate that an abstract method must always be a different in the supplies making overriding is compulsory is done using the keyword abstract.

Syntax:

Abstract returntype method name(paranthesis);

4. we cannot use abstract class into initialize object directly.

5. we cannot declare abstract constructor or abstract static methods

Abstract class:-

1. An abstract class is a class that contains one or more abstract methods.

2. An abstract class cannot be Initialize.

Syntax:-

Abstract class classname

{

Variable declaration;

Non-abstract method;

Abstract method;

}

Eg:

Abstract class shape

{

Int a,b;

Void display();

Abstract int Draw(int x, int y);

}

3.3

INTERFACE

1.Explain about interface .(or)How to implement interface in java.

A. **Interface:-**

1. An interface is basically a kind of class.

2. An interface contains only abstract methods & final fields (variables).

3. All methods in the interface are abstract method

4. These abstract methods do not specify any code to implement that variables &methods contains only constants.

Defining An interface:-

Syntax:-

Interface interfacename

{

Variable declaration;

Method declaration;

}

In the above syntax , interface is a keyword and interface name is any java valid identifier.

Variable & declaration:-

All variables declared as constants in interface.

Syntax:

Static final datatype variablename =value;

Static final int X= 24;

Static final float pi = 3.14f ;

Method declaration:-

Method declaration contains list of methods and anybody statement.

Syntax :- return type method name (parameter list);

Eg:- float computer(float x, float y);

Example program for interface:-

```
Interface area

{

Static final float pi= 3.14 f;

Float compute(float x, float y);

}
```

Extending interface:-

→An interface can also be extended.

→An interface can be sub interfaced from other interfaces.

→The new sub interfaces will inherit all the properties of super interface. This is done by using an “extends” keyword.

Syntax:-

```
Interface sub interface extends super interface

{

Body of sub interface;

}

Eg; interface A

{

Static find float pi = 3.14f;

}

Interface B extends A

{
```

Float compute (float x, floaty);

}.

Implementing interface:-

Interface are used as super class , whose properties are inherited by class.

Syntax:-

Class class name implements interfacename

{

Body of class;

}

Example :-

Class rectangle implements area

{

Float compute (floatx, float y)

{

Return(x*y);

}

}

Accessing interface variable: -

Interface can be used to declare a set of constants that can be used in different classes.

Example program:-

Interface area

{

Static final float pi=3.14 ;

Float compute (float x, float y);

```
}
Class rectangle implements area
{
Public float compute(float x, float y);
{
Return(x*y);
}
}
Class  circle implements area
{
Public float compute(float x, float y)
{
Return(pi*x*y);
}
}
Class inter
{
Public static void main(string args [])
{
Rectangle rect=new rectangle ();
Circle cir=new circle();
area ar;
ar=rect;
ar=cir;
```

s.o.p(“Area of rectangle is ”+area. Compute (10,20));

s.o.p(“Area of circle is”+ar. Compute(10,0));

}

}

Output:

Javac inter.Java

Java inter4

Area of rectangle is 628.0

Area of circle is 0.0

3.2

ARRAYS, STRINGS, VECTORS WRAPPER CLASS

1 . Explain about arrays in java?(or)How to declare and initialization of arrays in java.

A .**Array:-** An array is a variable that can store multiple values of same data type where as an ordinary data variable can store a single value at a time.

Types of arrays:-

Dimensional array:-

1.An array in which list of elements are stored in a continuous memory locations and access only one subscript.

Declaration of I D array:

Syntax: -

Type arrayname[ ];

Or

Type [ ] arrayname;

In the above syntax type is datatype, arrayname is name of array remember we do not enter the size of the array in the declaration.

Create memory locations:-

After declaring an array we need to create it in the memory Java allows us to create arrays using new operator only.

Syntax:-

Array name = new type [size];

Eg: a= new int [5];

The two statements may be combined as follows.

Syntax:-

Type arrayname[ ]=new type [size];

Eg: int a [ ]= new int [5];

The total size of array is 5 x 4 = 20 bytes.

This is because each integer element required 4 bytes for storage.

Initialization of 1D array:

Assigning value to an array when it is created is known as initialization.

Example:

Int a[] = {5,4,2,6,1};

In the above example five elements are stored in array a the array element are stored in continuous memory locations.

A[0] is assigned value 5. Similarly ,

a [1] is assigned value 4, a[2] is assigned the value 2.

*a[3] =6

a [4]= 1.

a[0] a [1] a[2] a[3] a [4]

5 4 2 6 1

Array length:-

Java we can find the length of the array "a" using "a .length".

Example:-

```
Int n=a.length

For (i=0;i<n;i++)

System.out.println("  "+ a[] );
```

Two dimensional array :-

An array in which list of elements are stored in same rows and columns and accessed by using two subscripts.

Declaration:

Syntax:-

```
Type array name [] [];

Or

type [ ] [ ] arrayname;
```

Here, type is a data type array name is the name of the array .Remember we do not have the size of the error in the declaration.

create memory location:-

After declaring an array we need to create it in the memory Java allows us to create arrays using "new" operator only.

Syntax:-

```
Array name = new type [row size] [column size];
```

Example

```
A =new int [5][3];
```

The total size of the array is 5 x 3 x 4 =60 bytes.

This is because each integer element require 4 bytes for storage

Initialization of a 2D array:-

Assigning values to an array when it is created is known as initialization.

Syntax :-

Type arrayname [ ] [ ] = {list of values of 1 st array}, {list of values of 2nd row },…{list of values of last row};

EXAMPLE;-

Int a [ ] [ ]= 5,6,3,2,4

Subscripts can also reflected in for loops that the array elements.

Example :

```
For(i=0 ; i<r ; j++)

For

System.out.println(“  ”+a[i][j]);

}

}
```

STRINGS

1 .Explain about strings in java (or) what is string? Write about various string methods in Java.

A String:- String represents sequence characters in Java, strings are class objects and implemented using two classes namely “string & string buffer”.

A java string is an initialized of the string class”. A java string is not a character array and is not terminated with null.

Syntax -1:

String Stringname;

Syntax -2:

String name=new string(“string”):

Eg: string a;

Eg:

A=new string(“Hari”);

The two statements may be combined as follows .

Syntax

String stringname = new string (“string”);

Eg;

String a=new string(“Hari”);

It is possible to get the length of string using the “length method ” of the stirng class

Int m = a legth();

String array : we can also create and use arrays that contain strings.

Syntax :

Stirng stringname [] =new string [size];

Eg: string a[ ] =new string [3];

We will create an array of size ‘3’ to add “3” string constants.

String method.

The string class defines a number of methods that allows us to perform a variety of string manipulation tasks.

Method	Task performed
S2 =S1.toLowerCase();	convert the string S1 to all Lower case Ex: String S1=new string (“Java”); S2= S1 to Lower case();

Output; Java

S2 =S1.to uppercase();	convert the string S1 to all upper Case
S2=S1 to replace('x','y');	replace the character of 'x' with 'y'.
S2=S1.trim()	Remove white spaces at the beginning& end of the string.
S1.equals (S2);	Return "true", if S1==S2.
S1.length()	gives the length of string 'S1'.
S1.charAt(n);	gives nth position of character 'S1'
S1.comapre to(S2);	return "_" , if S1<S2, '+' if S1>S2 & "0" if S1=S2.
S1. Concat(S2);	adding S1 & S2.
S1.substring (n);	gives substring starting from nth character upto mth character
S1.Substring (n,m);	gives substring starting form nth character upto mth character.

string buffer class:-

String buffer class is similar to string class while string creates strings of fixed length,

String buffer strings of flexible length.

That can be modified in terms of both length and content.

We can insert character in the middle of the string are appeared string at the end.

Syntax :

stringbuffer string name =new stringbuffer("String").

Eg: string buffer a= new string buffer ("Java");

Method	Task performed
S1.chart At(n,'x');	Modifies the nthy character to 'x'.

S1append(S2);	apped the string S2 to S1 at the end.
S1.insert(n,S2;)	insert the string S2 at the position ‘n’ of the strintg ‘s’.

2.what is vector explain various methods in vector class?

A.Vector:-

* vector class can be used to create a dynamic array known as vector.

* That can hold objects of any type and any number.

→vectors are created like array as follows:-

vectors contains “Java.util” package.

Syntax:-

vector vector name=new vectore(); //(declaring without S3//

2. vector vectorname = new vector (size); // (declaring without size)//.

methods in vector class:-

add element (item):-

it is used to add the specified item to the vector at the end

eg:- V.add element(item);

insert element :- it is used to add the specified item at nth position to the vector at the end.

Eg:- insert Element At (“Java”,2);

Element At(n):-

it is used to get the name of the object

eg:- element At (2)

size:-

it is used to find the number of objects in the vector.

Eg :- V.size();

Remove element (item):-

it is used to remove the specified item from the vector.

eg :- remove Element ("Java");

Remove all elements At (n):-

It is used to remove the element stored in the nth position of the vector .

Eg :- remove Element (3);

Remove All elements():-

It is used to remove all the elements from the vector.

Eg :- remove All elements();

Copy into:-

It is used to copy all the objects from the vector to array

Advantages;-

1 it is the convenient to use vector to store the objects.

2 a vector can be used to store a list of objects that may vary in size.

3 we can add and delete objects from the list as and when required.

The major constraint in using vectors is that we cannot directly store simple data type in your data we can only store objects therefore we need to convert simple data type to objects this can be done using the wrapper class.

3. Write about wrapper class.

A .Vectors cannot handle primitive data types like int ,float ,long, char and double. primitive data types may be converted into objects types by using wrapper class. Wrapper classes

Contained in "Java.lang" package

Wrapper classes for converting simple types:

Data type	Wrapper classes
Boolean	Boolean

Char	Character
Double	Double
Float	Float
Int	Integer
long	Long**:-**

Converting primitive numbers to object numbers using constructor method

Constructor calling	**Conversion action**
Integer Int val= new Integer(i);	Primitive integer to integer object
Float Float val=new Float (f);	Primitive float to float object
Double Double Val= new Double (d);	Primitive double to double object
Long Long val = new Long();	Primitive long to long object

Converting object numbers to primitive numbers using type value() method:-

Method calling	**Conversation action**
Int I = Int val.int value();	Object to primitive integer
Float f=Float val.float value();	Object to primitive float
Double d=Double val.double value();	Object to primitive double
Long l = Long val.long value();	Object to primitive long.

Converting number to string using to string() method.

Mehtod calling	**Conversation action**
Str=integer.to string(i);	Primitive integer to string
Str=Float.to string(f);	Primitive float to string
Str=double.to String	Primitive double to string
Str=Long.to String	Primitive long to string

Convert numeric string to primitive numbers :-

Method calling	Conversation action
Int I = Integer.parse int(Str)	Convert string into primitive integer
Float f=Float.parse Float(Str)	Convert string into primitive float
Long l= long.parse long(str)	Convert string into primitive long

UNIT -4

4.2 MANAGING ERRORS AND EXCEPTIONS.

1.Write about write about types of errors with examples.

A. **Errors:-**

An Error may produce an incorrect output or may terminate the executive of the Program abnormally. Errors may be classified into two categories.

1. Compile time.

2. Run time.

Compile time:-

All Syntax errors will detect and therefore these errors are known as compile time errors Whenever the company displays an error it will not create the class file Java compiler does a Nice job of telling us where the errors are in the program most of the compile time errors are due to type mistakes .

Some of the Compile time errors are:-

1. Missing semi Columns(;)

2. Missing brackets in classes and methods.

3. Misspelling of Identifiers and keywords.

4. Use of equal to in place of double equal to operator.

5. Missing double quotes in string.

Eg :-

(compile time)

```
class error
{
public static void main(string args [ ])
{
Int x,y //semi column missing
Void get()
{
X=10;
Y=20;
} // missing "}" barces.
}
}
```

Runtime errors:- A program may compile successfully creating the class file but may not run properly such programs may produce of wrong results due to wrong logic.

1. Some of the runtime errors are dividing an integer by zero.

2. Accessing an element that is out of the bounds of an array.

(Runtime)

```
Class error
{
Public static void main(args [ ])
{
Int x, y;
```

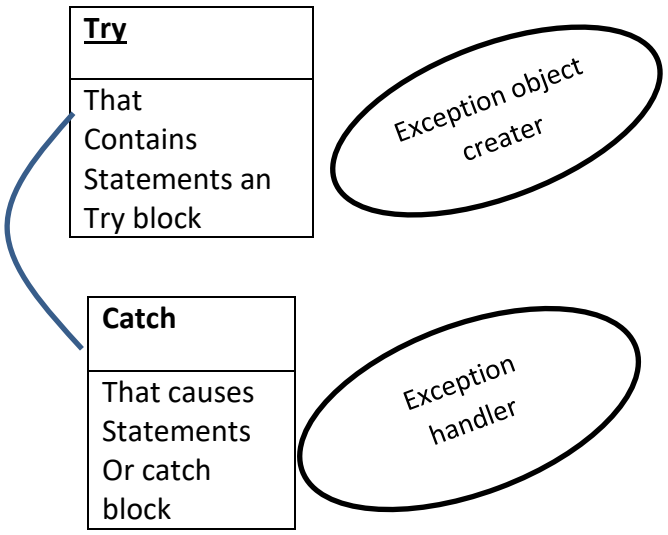
```
X=10;

Y=0;

Int c =x/y;  //(that causes an array at runtime)

System.out.println("c"+c);

}

}
```

2. What is exception explain exception handling mechanism and its types of exception.

A . **Exception** :-an exception is an event that occurs during the execution of program that disturbs in a normal flow of execution when an error occur within a method the method creates an object called an "Exception object" contents information about the error including its type and state of the program.

→ creating an exhibition object and handling it to the runtime system is called throwing an exception.

Exception handling:-



If the exception object is not caught and handled properly , the interpreter will display error Message and will terminate the program. If we want the program to continue with execution of remaining code then we should try to catch the exception object thrown by the error

Condition and display an appropriate Message for taking corrective actions.

This is known as exception handling.

Error handling code performs the following tasks.

1. Find the problem hit the exception

2. Inform that an error has occurred (throws the exception).

3. Receive the error information catch the exception.

4. Take corrective actions (Handle the exception).

The error handling code basically consists of two segments:-

1. To detect errors and throw exceptions.
2. To take exceptions and to take appropriate actions.

Synatx:-

```
Try
{
Statements ; //generate an exception.
}
Catch (exception -type e)
{
Statements ; // process  the exception
}
```

Java uses as keyword “try”. To preface a block of code i.e likely to cause an error condition and throw an exception.

A catch block defined by the key word catch;

Catch catches the exception thrown by the try block and handles it approximately. The catch block is added immediately after the try block.

Example:-

```
Class error 1
{
Public static void main(String args  [ ])
{
Int a= 10;
Int b= 5;
Int c= 5;
Int x,y;
Try
{
X= a/(b.c);
}
Catch(Automatic exception e)
{
System.out.println(“Division by zero”);
}
Y= a/(b+c);
System.out.println(“y=”+y);
}
}
```

Output:-

Division by zero.

Exception types:-

There are two types of exceptions

1 Checked exception:-

These exceptions are handled in the program.

Itself with the help of “try” and “catch” blocks.

These are extended from the

Java.lang.exception class.

2 Unchecked exceptions:-

These exceptions are not handled in the program but the JVM handles such exceptions. These

are extended from the “Java.Lang.Runtime” exceptions class.

Some of the exception types are:-

Exception type	Cause of an exception
1 Arthematic Exception	Caused by mathematical errors such as division by zero
2 Array Index out of Bounds Exception	Caused by bad array indexes
3 Number Format Exception	Caused when convert between string &numbers
4 Out of memory Exception	Caused when there is no enough memory to allocate new object
5 Null Pointer Exception	Caused by referring a null object
6 File Not Found Exception	Caused by attempt to access a non-existing file.
7 Array store exception	Caused when a program try to store the wrong data in an array
8 IO Exception	Caused by general input (or) output failures
9Security Exception	Caused when an applet try to perform an action that is not allowed by the browser

3 Explain multiple catch statements or multiple catch.

Multiple catch statement:-

Syntax:-

Try

{

Statements;

}

Catch(Exception type 1 e)

{

Statements;

}

Catch(Exception – type2 - e)

{

Statements;

}

Catch (Exception-type n – e)

{

Statements;

}

Process:-

When an expression in a try block is generated, the Java creates the multiple catch statements like classes in a switch statement.

The first statement whose parameter matches with the exception and remaining statements will be skipped.

Eg :- program:-

Class example

{

Public static void main (String args)

```
{
Int a[] = {10,5};
Int b=5;
Try
{
Int x=(a1/b-a2);
}
Catch (Arthematic Exception e)
{
 System.out.println("Division by zero");
}
Catch(Array Index out of bounds exception e )
{
System.out.println("Bad Index error");
}
Catch Array store Exception  e)
{
Sytem.out.println("wrong data");
}
}
}
```

4 Explain finally statements?

A . Finally statements :- Java supports another statements known as finally sattements that can be sued to handle an exception that is not catch any of the previous catch statements.

→Finally block can be used to handle any exception generated within a try block.It may be added immediately after the try block or after the catch block as follows:-

Synatx 1

```
Try
{

Statements;
}
Finally()
{

Statements;
}
}
```

Sytax 2

```
try
 {
  Statements;
catch(Exception type e)
 {
 statements;
 }
 fianlly()
 {
 statements;
```

4.1

MULTI-THREADED PROGRAMMING.

1 Write about multithreading program.

A . Multithreading :- where program is divided into two or more sub program which can be implemented at the same time.

Thread :- It is similar to a program that has a single flow of control has a beginning a body and an end an executable commands.

Java is support for multithreading i.e;

Java enables us to use multiple flow of control in developing programs.

In the Program contains multiple flow of control is known as multithreading Java program will have four Threads ,1 main and other 3 sub the main thread is contained in the main model and restart the other three methods.

Threads in Java subprograms of a main thread and share the same memory space they are known as light weight thread or “light weight process.”

Create a thread:- Threads are implemented with the form of an object that contained a method called “run()”. This run() is the heart of the thread.

Syntax:-

Public void main()

{

Sattements;

}

The Run() should be invoked by an object of the concern thread the Run() can be executed by using “start()” method.

A new thread can be created into two ways .

By creating a thread class:-

Defining a class that extends “thread” class and override its Run method.

By converting a class to a thread:-

Define a class that implements runnable interface the Rebel interface has only one run method that can be executed by the thread

Extending the thread class.

Defining a class by extending using “Java. Lang.Thread”.

*Define the class as extending the thread class.

Syntax:-

Class my thread extends thread.

{

Statements;

}

- Implement the run() method i.e responsible for excluding the sequence of program that the thread will execute.

Eg :-

public void main run ()

{

--------;

}

- Creating a thread object and call the "start()"

Eg :- My thread t = new My thread();

t.start();

Stopping a thread

when we want to stop a thread from running state then call "stop method".

Eg :- t.Stop();

→This statement causes the third to move to dead state.

Blocking a thread :-

A thread can also be temporarily suspended or blocked from entering into a runnable are subsequent running state by using following 3 thread methods.

1 Sleep() → Blocked for a specific time.

2 suspended () → blocked for further order.

3 wait() → Blocked until certain condition.

* If the thread is blocked with suspends then thread is invoked by resume

**If that red is the Block with the weight then thread is invoked by notify

2 Explain life cycle of a thread or a thread life cycle?

A . Thread :-

It is similar to a program that has a single flow of control has a beginning a body and an end an executable commands.

life cycle of threads:-

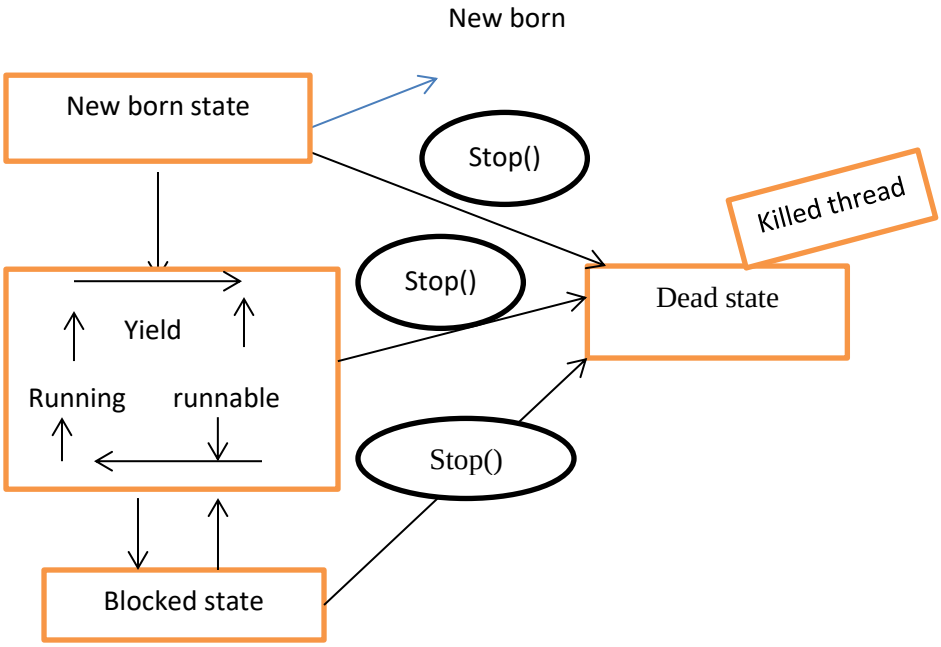
During the lifetime of a thread it contains five steps, they are:

1 New born state

2 Runnable state running state

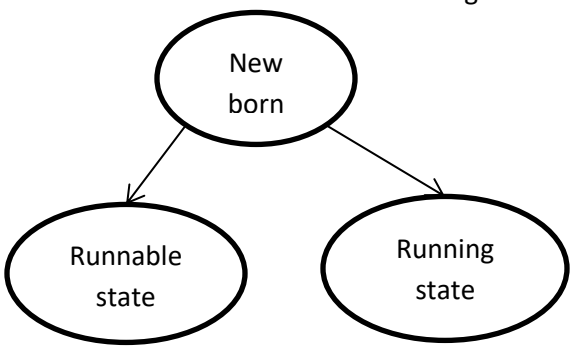
3 Blocked state

4 Dead state



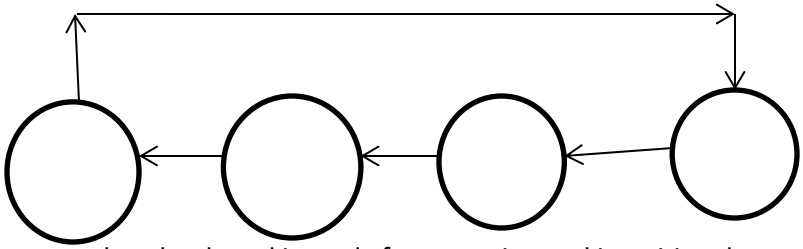
1 **New born state:-** when we create a thread object born and set to be in newborn

the thread is not scheduled for running at this state we can do. *scheduled it for running using “start()”.



* kill it using stop method.

Runnable state:-



The Runnable state means that the thread is ready for execution and is waiting the availability of the processor if all the Threads have equal priority then they are given time slots for execution in a round Robin fashion. This process of assigning time to threads is known as “time slicing”.

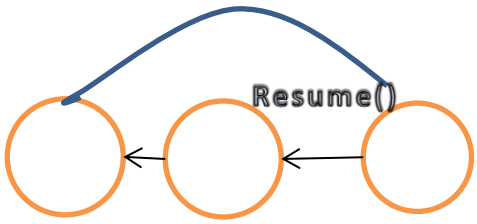
Yield is means that the current thread is willing to give another thread.

Running state:-

Running statements that the processor has give its time to the trade for its execution around credit control in one of the following situations.

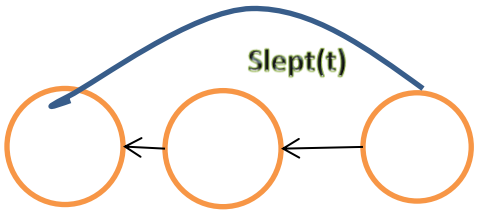
A) It has been suspended using suspended a suspended thread can be released by using the resume().

Suspend



Running thread runnable thread suspend

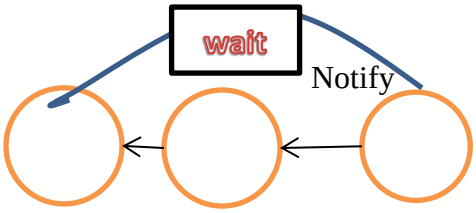
b) It has been made to sleep with can put at thread to sleep for a specified time period using the method sleep time where time is in milliseconds



Running runnable sleeping.

Thread thread

c.It has been made to wait until some event occurs this is done using white this thread can be scheduled to run away when using notify



Blocked state:-

- A thread is said to be blocked when it is prevented from entering into the runnable state and sequentially the running state. This happens when the thread is suspended are waiting in order to satisfy the certain requirements.

Dead state:-

- Every thread has a life cycle.
- running and life when it has completed executing its Run
- It is a natural death
- we can kill it by sending the stop message to it at any state

3 write about thread priority.

A. Thread priority:-

In java is thread is assigned a priority which affects the order in which its scheduled for running.

The thread of same priority are given. Equal treatment by Java scheduler and therefore they shared the processor on a Round Robin fashion Java permits as to priority of a thread u sing the set priority() is as follows.

Sytax:-

Thread Name.set priority(int number);

- The int numbers is an integer value to which the threads priority is set. The thread class defines several priority constants

MIN -- PRIORITY=0

NORM – PRIORITY= S

MAX – PRIORITY=10

- The int numbers may resume one of these constants (or) any value between '0' to '10'

(Note The default setting is NORM – PRIORITY)

The multiple threads are ready for execution the java system chooses the highest priority thread and execute it.

1 it stop running at the end of run().
2 it is made to sleep using sleep().
3 It is told to wait using wait().

4 Explain the thread exception?

A. When the thread is blocked with sleep () is enclosed in a try block and followed by a catch block.

* This is necessary because the sleep() throw an exception we should be kept.

* If we fail to catch the exception program will not compile.

* Java runtime system will throw "illegal thread state exception" whenever we attempt to invoke a method that a thread cannot handle in the given statement.

*When we call a thread method that is likely to throw an exception handler to catch it.

The catch statement may take one of the following forms.

```
Catch(Thread Death  e)
{
Statements; // Killed thread.
}
Catch (interrupted exception e)
{
Statements; // cannot handle it in the current statement.
Catch (illegal Argument exception  e
{
Statements;   // Illegal methods)
}
catch( exception  e)
{
Statements;  // Any other.
}
```

5 write about synchronization?

A . synchronization

synchronization generally means sharing data between multiple threads.

One thread main try to read record from a file while another is still waiting for same file.

Depending on the situation Java and able to overcome this problem using a technique known as synchronization.

In case of Java the keyword synchronized help to solve such problems by keeping a watch on such locations.

Eg :-
Synchronization void update()
{
Statements; // code here is synchronized.
When we declare a method synchronized. Java creates a “monitor” and hands it over to the thread that calls the method first time.
Eg:- synchronized (lock object)
{
Statements;

} .

6 Explain runnable interface (or) implement their runnable interface

A Runnable Interface declare the run () that is required for implementing threads in your programs.

1 Declare the classes implementing the runnable interface.

2 Implementing Run().

3 Create a thread by defining an object that is initialized from the runnable class as target of the thread

4 Call the thread start to run the thread.

Eg
```
import.java.lang.Thread;
Class x implements Runnable  // step 1
{
```

```
Public void run()  // step 2
{
For(int i=1; i<=10; i++)
{
System.out.println("Thread x ; i" +i);
}
System.out.println("Exit from x ");
}
}
Class Runnable Test
{
Public static void main(string args [])
{
X   r= new x();
Thread Thread x=new Thread(r):
Thread X.start(t);
System.out.println("end of main method is");
}
}
```

5.2 packages

1 what is package explain about different types of packages?

A package:- package is a container in which a variety of classes or interfaces are grouped together. The grouping is done according to functionality.

Advantages:-

The classes contained in the packages of other programs can be easily reused.

In the packages classes can be unique two classes in two different packages can have the same name.

packages provide the way to hide the classes.

packages as a provider for separately design for coding.

Types of packages:-

1 Java packages are classified into two types they are :-

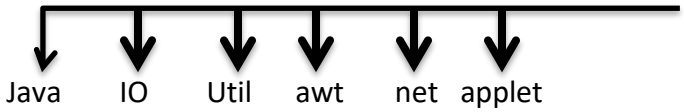
Java API package (Application Programming Interface) predefined package.

2 User defined package.

Java API package:-

It provides a large number of classes grouped into different packages according to functionality packages are organized in hierarchical.

Frequently used API packages:-



package name content:-

1 java.lang	it includes classes for primitive type, strings ,math functions , Threads and exceptions
2Java.util	it includes utility classes such as vectors Random numbers data are type etc
3 Java.io	they provide facilities for input and output of a data
4java.awt	it implementing graphical user interface the (GUI) then includes classes for Windows menus
5 java.net	they include classes for communication with the local computers as well as internet servers
6 java.applet	classes for creating and implementing applets

Using API package:-

There are two ways to access classes Store package they are :-

The first approach:-

The approach was the fully qualified class name the “fully qualified class name” The fuly qualified class name consist of Java package name and classname separately by dots representing different levels.

Process :-

→use the package name in which the required class is contained.

→use the class name in which the required method is contained.

Syntax:-

Java .packagename. classname;

Eg:- Java.awt.colour;

In the above example “awt” is the package of the Java package and colour is the class of the WT package.

The second approach:-

This approach is just a import statement this statement must be written at the top of the program it consists of import keyword package name and class name along with dot.

Process:-

*uses the import keyword.

*uses the package name in which the required class is contained.

*use the class name in which the required materials contained.

Syntax:-

Import Java.package name .class name;

Example:-

1 import Java.out.colour;

2 import java.awt .*;

User defined package:-

packages that are design and understand that by the user are called used at different packages.

Naming action :-

- Packages can be named using the standard java naming rules.
- All packagename begin with lower case letter.
- All method names begin with lower case letter.

Syntax:-

Java. Packagename.calssname.method name;

Eg :- java.lang.Math.sqrt(x);

2 what is user defined package how to create and access user defined package?

User defined package:-

package that are designed and created by the user are called user defined

packages

creating a package:-

To create a package must first deploy area package using the package keyword

then we must define a class using public

Declaring a package:-

Synatx:-

```
Package packagename;
Public class classname;
{
Statements; // body
}
Eg ;- package Student;
Public class Sum
{
Public int add(int x, int y)
{
Return(x+y);
}
}
```

Steps for creating packages:-

*Decide the name of the package.

*Creating a sub directory with this name under the directory where main source files are stored

D:/MCCs /> Md student.

D:/MCCs/> Cd student

D:/MCCs/student/>edit

*Declare the package at the beginning of the file using the following syntax.

Synatx :-

Package packagename;

*The name of the sub directory must be the name of the package exactly.

*Define the class that is to be put in the package and declare it using “public”.

Eg :
```
package student;
Public class Sum
{
Publc int add(int x, int y)
{
Return(x+y);
}
}
```
*Save the source file by c;ass name, “sum.java”.

*Switch to sub directory created earlier and compile the source file when completed, the package contain “sum.class file” of the source file.

Accessing the package (run uses):-

The import statement is used to use t=a package . The general form of using package is

Syntax

Import package name .calssname;

Eg : import student.sum;

Here , student id the name of the package and sum is the name of the class in it.

Eg :-

```
Import student.Sum;
Calss Test
{
Public static void main(String args  )
{
Sum s= new Sum()””;
Int n=S.add(2,3);
System.out.println(“sum is”+n);
```

```
}
}
```

Adding a class to a package:-

we can also add in new class two packages

suppose we want to add another class Sab to the student package then follow the steps

* Define their class and make it public.

*place the package statement before the class definition as follows.

Eg:

```
package student ;
Public class Sub
{
Public int Sub(int x , int y)
Return(x-y);
}
}
```

*store this as sub.java file under the directory student

*compare Sab dot Java file. this will create a subclass file under place it in the directory student

3 How to hide the class in packages hiding classes?

Hiding classes:-

When we import packages using "*" all public classes are imported.

If we want to hide a particular class we must not use "public" in the class definition

Syntax:-

```
Package packagename;
Public class classname1
{
Body of the classname1;
}
Class classname2
{
Body of the classname2;
```

}

Here, class2 is not declared in "public" , that class is hidden from out side of the package.

Eg :-

```
Package p1;
Public class A
{
Public void dislay()
{
System.outprintln("This is a class B");
}
}
```

4 Write about static import?

Static import:-

Static import is another feature introducedwith J2SE 5.0 release.

This feature eliminates the need of qualifying a static member with class name static import declaration is similar to that of import .

We can use the import statement to import class .From packages and use them without qualifying the package.

Similarly, we can use the static import statement to import static members from classes and use them without qualifying the class name

Syntax:- import static packagename.classfile.staticmenbername;

Eg :-

```
Import static java.lang.math.*;
Public class math op
{
Double area = pi*r*r;
Sytem.out.println("Area"+area);
}
Public static void main (String args[])
{
Mathop obj = new mathop;
Obj.circle(2,3);
}
}
```