25. Write a Java Program to create and start a thread.

class th extends Thread

{

public void run()

{

System.out.println("Thread is running");

}

public static void main(String args[])

{

th t=new th();

t.start();

}

}

26. Write a Java Program to demonstrate the working of Thread Priority

class tpri extends Thread

{

public void run()

{

System.out.println("Inside Run method");

}

public static void main(String args[])

{

tpri t1=new tpri();

tpri t2=new tpri();

System.out.println("Default thread priorities");

System.out.println("Thread 1 priority"+t1.getPriority());

System.out.println("Thread 2 priority"+t2.getPriority());

System.out.println("Max and Min thread priorities");

t1.setPriority(MAX\_PRIORITY);

t2.setPriority(MIN\_PRIORITY);

System.out.println("Thread 1 priority"+t1.getPriority());

System.out.println("Thread 2 priority"+t2.getPriority());

t1.setPriority(2);

t2.setPriority(9);

System.out.println("After setting priority");

System.out.println("Thread 1 priority"+t1.getPriority());

System.out.println("Thread 2 priority"+t2.getPriority());

System.out.println("Current runing thread is "+currentThread().getName());

}

}

27. Write a Java Program to demonstrate working of join () method in threads.

class join extends Thread

{

public void run()

{

for(int i=0;i<3;i++)

{

try

{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e);

}

System.out.println(i);

}

}

public static void main(String args[])

{

join j1=new join();

join j2=new join();

join j3=new join();

j1.start();

try

{

j1.join();

}

catch(Exception e)

{

System.out.println(e);

}

j2.start();

try

{

j2.join();

}

catch(Exception e)

{

System.out.println(e);

}

j3.start();

try

{

j3.join();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

28. Write a Java Program to demonstrate the working of Thread Synchronization.

class table

{

void printtable(int n)

{

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

{

System.out.println(n\*i);

try

{

Thread.sleep(1000);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

class mythread1 extends Thread

{

table t;

mythread1(table t)

{

this.t=t;

}

public void run()

{

t.printtable(2);

}

}

class mythread2 extends Thread

{

table t;

mythread2(table t)

{

this.t=t;

}

public void run()

{

t.printtable(5);

}

}

class test{

public static void main(String args[]){

table obj = new table();//only one object

mythread1 t1=new mythread1(obj);

mythread2 t2=new mythread2(obj);

t1.start();

t2.start();

}

}

29. Write a Java collections program to demonstrate ArrayList interface.

import java.util.ArrayList;

class arrl {

public static void main(String[] args) {

ArrayList<String> language = new ArrayList<>();

// add elements in the array list

language.add("Java");

language.add("RDBMS");

language.add("OS");

System.out.println("ArrayList: " + language);

// remove element from index 2

String str = language.remove(2);

System.out.println("Updated ArrayList: " + language);

System.out.println("Removed Element: " + str);

}

}

30. Write Java collections programs to demonstrate LinkedList interface.

import java.util.\*;

class al

{

public static void main(String args[])

{

LinkedList <String> list=new LinkedList <String>();

list.add("C");

list.add("C++");

list.add("Web Programming");

Iterator itr=list.iterator();

while(itr.hasNext())

{

System.out.println(itr.next());

}

while(itr.hasNext())

{

System.out.println(itr.next());

}

}

}