Name: Emp ID: Date:17.08.2020

Duration: 30 Min.

# Question 1:

What are the ways of creating threads in JAVA?

1. Extending the java.lang.Thread class
2. Extending the java.lang.Runnable class
3. Implementing the java.lang.Runnable interface
4. Implementing the java.lang.Thread interface

Options :

# (i) & (iv)

* 1. (ii) & (iii)
  2. (i) & (iii)
  3. (iv) & (ii)

# Question 2:

Which method is used to bring the Thread from RUNNING to RUNNABLE state?

Options :

1. Sleep()
2. Wait()
3. Notify()

# Yield()

**Question 3 :**

Which of the following is/are TRUE about Multithreading?

1. Threads are lightweight because they share registers & stacks.
2. Threads are memory efficient as they share the same address space.
3. Threads have less context to save than a process.
4. Threads have separate *code*, *heap area*, *files.*
5. Improves performance of an application.

Options :

A. (i),(ii),(iii),(v)

B. (ii),(iii),(iv),(v)

# C. (ii),(iii),(v)

D. All the options.

# Question 4 :

With reference to the Thread States in JAVA which of the following is/are TRUE?

1. RUNNABLE  wait()  WAITING
2. NEW BORN  start()  RUNNABLE
3. RUNNING  yield()  RUNNABLE
4. WAITING  notifyAll()  RUNNABLE
5. RUNNING  I/O Request  BLOCKED

Options :

A. (i),(ii),(iii),(v)

# B. (ii),(iii),(iv),(v)

C. (ii),(iv),(v)

D. All the options

# Question 5:

Which method must be defined by a class implementing the java.lang.Runnable interface?

1. void run()

# public void run()

1. public void start()
2. void run(int priority)

E public void run(int priority)

F. public void start(int priority)

# Question 6:

The output of the following code is public class ReTest

{

public static void main(String argv[])

{

TestThread tt1 = new TestThread("One"); tt1.start();

TestThread tt2 = new TestThread("Two"); tt2.start();

}

}

class TestThread extends Thread

{

String sTname=""; TestThread(String s)

{

sTname = s;

}

public void run()

{

for(int i =0; i < 2 ; i++)

{

try

{

sleep(1000);

}catch(InterruptedException e){}

System.out.println(sTname);

}

}

}

1. Compile time error, class ReTest does not i
2. Output of One One Two Two
3. Output of One Two One Two

# Compilation but no output at runtime

**Question 7:**

What can cause a thread to stop executing under some circumstances

a The program exits via a call to System.exit(0); b Another thread is given a higher priority

# A call to the thread's stop method.

1. A call to the halt method of the Thread class? e A call to the Thread.reStart method

# Question 8:

What will happen when you attempt to compile and run the following code?

class TSamp extends Thread{ public native String getTime();

}

public class Multi implements Runnable { boolean bStop;

public static void main(String argv[]){ Multi m = new Multi();

m.go();

}

public void go(){

TSamp ts = new TSamp(this); ts.start(); bStop=true;

}

public void run(){ if(bStop==true){

return;

}

System.out.println("running");

}

}

Options:

a Compilation, but output at runtime cannot be exactly determined b Compilation and output of "running"

1. Compilation but no output at runtime

# Compile time error

**Question 9:**

The following block of code creates a Thread using a Runnable target:

Runnable target = new MyRunnable(); Thread myThread = new Thread(target);

Whlch of the following classes can be used to create the target, so that the preceeding code compiles correctly?

a).public class MyRunnable extends Runnable { public void run(){}} b).public class MyRunnable implements Object{ public void run(){}} c).public class MyRunnable implements Runnable { public void run() {}} d).public class MyRunnable extends Runnable { void run() {}}

# Question 10:

The parameter for sleep method of thread class is in

* 1. Seconds
  2. Minutes

# Milliseconds

* 1. Micorseconds

# Question 11:

Thread t = new Thread();

The following code takes the thread in which state

1. Runnable
2. New
3. Running

# None

**Question 12**.

Who maintains the life cycle of the servlet?

# Servlet container

1. the client
2. Both a and b

**Question 13**

.Which of the following method is called when the Servlet is first loaded?

1. initialize() method

# init() method

1. Both of the above

**Question 14.**

Which of the following is a feature of Servlet?

1. Supports Multithreading
2. Platform independent
3. Protocol independent

# All of the above

**Question 15**.

HTTP header fields include

1. Content-length
2. Cookies
3. Type of request - POST or GET

# All the above

**Question 16.**

What will happen when the code is compiled and executed?

import javax. servlet.\*; import javax.servlet.http.\*; import java.io.\*;

public class TestServlet extends HttpServlet

{

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

response.setContentType("text/html"); PrintWriter out=response.getWriter(); out.println("<b> Servlet Exam"); out.close();

}

public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

doGet(request, response);

}

}

1. Compilation Error
2. Compiles fine but will not be executed. Displays error.

# Displays output as Servlet Exam in HTML bold text

1. Displays output as <b> Servlet Exam

**Question 17.**

Servlets are better than CGI because

1. Servlets are platform independant
2. Servlets are scalable
   1. 1 is true
   2. 2 is true

# Both are true

**Question 18**

Consider the following servlet import javax.servlet.\*;

import javax.servlet.http.\*; import java.io.\*;

public class MyServlet extends HttpServlet { int var;

public void init() throws ServletException { var=0;

} ....

}

For every browser request, a new instance of the servlet is created by the server

1. True

# False

**Question 19.**

Which of the following is not a server-side technology?

1. Servlets
2. Java Server Pages

# DHTML

1. CGI

**Question 20:**

HTTP is stateful protocol

a).True

# b).False