|  |  |
| --- | --- |
| Course- BTech | Type- Core |
| Course Code- **CSET109** | Course Name- **Object Oriented Programming Using Java** |
| Year- First | Semester- Even Batch- BTech 2nd Semester |

**Tutorial-11**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tutorial No.** | **Name** | **CO1** | **CO2** | **CO3** |
| **11** | **Multithreading and File handling** | **--** |  | **--** |

**Objective:** The main objective of this tutorial is to learn about the multithreading and file handling concepts of Java language.

* 1. What will be the output of the following program? public class IsItCorrect {

public static void main(String[] args) throws InterruptedException { Runnable r = new Runnable() {

public void run() { try {

Thread.sleep(1);

} catch (InterruptedException e) { System.out.print("Interrupted,");

}

System.out.print("Run,");

}

};

Thread t = new Thread(); t.run(); System.out.print("Started,"); t.start();

System.out.print("Interrupting,"); t.interrupt(); System.out.print("Ended,");

}

}

Output: Started,Interrupting,Ended,

* 1. What will be the output of the following program? public class Test implements Runnable

{

public void run()

{

System.out.printf("Hello "); System.out.printf("World");

}

public static void main(String[] args)

{

Test obj = new Test();

Thread thread = new Thread(obj);

thread.start(); System.out.printf("World"); try

{

thread.join();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

System.out.println("for ");

}

}

Output: WorldHello Worldfor

* 1. What will be the output of the following Java program?

public class WaitTest

{

public static void main(String [] args)

{

System.out.print("1 "); synchronized(args)

{

System.out.print("2 "); try

{

args.wait(); /\* Line 11 \*/

}

catch(InterruptedException e){ }

}

System.out.print("3 ");

}

}

Output: 1 and 2 will be printed, but there will be no return from the wait call because no other thread will notify the main thread, so 3 will never be printed. It's frozen at line 7.

* 1. What will be the output of the program? public class OutputOfThread extends Thread {

private OutputOfThread() { setPriority(NORM\_PRIORITY); setPriority(MAX\_PRIORITY); setPriority(MIN\_PRIORITY);

}

public void run() {

System.out.println("Run = " + Thread.currentThread().getPriority() + Thread.currentThread().getPriority());

}

public static void main(String[] args) { OutputOfThread oft = new OutputOfThread();

System.out.println("Main = " + Thread.MAX\_PRIORITY + Thread.MIN\_PRIORITY + Thread.NORM\_PRIORITY);

oft.start();

}

}

Output: Main = 1015

Run = 11

* 1. What will be the output of the program? class MyThread extends Thread

{

MyThread()

{

System.out.print(" MyThread");

}

public void run()

{

System.out.print(" bar");

}

public void run(String s)

{

System.out.println(" baz");

}

}

public class TestThreads

{

public static void main (String [] args)

{

Thread t = new MyThread()

{

public void run()

{

System.out.println(" foo");

}

};

t.start();

}

}

Output: MyThread foo

* 1. What will be the output of this program?

public class HowMayTimes implements Runnable { int k = 0;

public HowMayTimes(int i) { k = i;

}

public static void main(String[] args) { new HowMayTimes(2).run();

new HowMayTimes(1).run();

}

public void run() {

for (int i = 0; i < k; i++) System.out.println("Simple Question");

}

}

Output:

Simple Question

Simple Question

Simple Question

* 1. What will be the output of the below program? public class NewQuestion {

public static void main(String[] args) { new VolatileTest("Merit").run(); new VolatileTest("Campus").run();

}

}

class VolatileTest extends Thread { private volatile int testValue; private volatile boolean ready; public VolatileTest(String str) {

super(str);

}

public void run() {

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

if (getName().equals("Merit")) { ready = true;

testValue = i;

System.out.println(getName() + ", " + ready + ", " + testValue);

}

if (getName().equals("Campus")) {

System.out.println(getName() + ", " + ready + ", " + testValue);

}

Thread.sleep(1000);

} catch (InterruptedException exception) { exception.printStackTrace();

}

}

}

}

Output:

Merit, true, 0

Merit, true, 1

Campus, false, 0

Campus, false, 0

* 1. What will be the output of this program? import java.io.\*;

class Chararrayinput

{

public static void main(String[] args)

{

String obj = "abcdef"; int length = obj.length(); char c[] = new char[length];

obj.getChars(0, length, c, 0);

CharArrayReader input1 = new CharArrayReader(c); CharArrayReader input2 = new CharArrayReader(c, 0, 3); int i;

try

{

while((i = input2.read()) != -1)

{

System.out.print((char)i);

}

}

catch (IOException e)

{

e.printStackTrace();

}

}

}

Output: abc

* 1. What will be the output of the following program? class s1 implements Runnable

{

int x = 0, y = 0;

int addX() {x++; return x;} int addY() {y++; return y;} public void run() {

for(int i = 0; i < 10; i++) System.out.println(addX() + " " + addY());

}

public static void main(String args[])

{

s1 run1 = new s1(); s1 run2 = new s1();

Thread t1 = new Thread(run1); Thread t2 = new Thread(run2); t1.start();

t2.start();

}

}

Output: 1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

1 1

2 2

3 3

4 4

5 5

6 6

7 7

8 8

9 9

10 10

* 1. What will be the output of the following program? public class Kashmir {

public static void main(String[] args) { try {

KanyaKumari kk = new KanyaKumari(); kk.run();

p('A');

kk.start();

kk.join();

p('H');

kk.run();

p('M');

kk.start();

p('B');

} catch (Exception ex) { p('I');

} p('R');

}

private static void p(char c) {

System.out.print(c);

}

}

class KanyaKumari extends Thread { String s = "KSMI";

static int i = 0; public void run() {

System.out.print(s.charAt(i++));

}

}

Output: KASHMMIR