

Q1: Гурван thread-г үүсгээд эхлүүлэх кодыг бичнэ үү.

```
public class TestThread
{
    /**Main method*/
    public static void main(String[] args)
    {
        // Create threads
        PrintChar printA = new PrintChar('a', 100);
        PrintChar printB = new PrintChar('b', 100);
        PrintNum print100 = new PrintNum(100);
        // Start threads
        print100.start();
        printA.start();
        printB.start();
    }
}
```

Q2: Дараах мөр кодуудыг гүйцээнэ үү.

```
class PrintNum extends Thread
class PrintChar extends Thread
```

Q3: PrintChar ба PrintNum классуудын run() функцүүдийг тодорхойлно уу.

// The thread class for printing a specified character

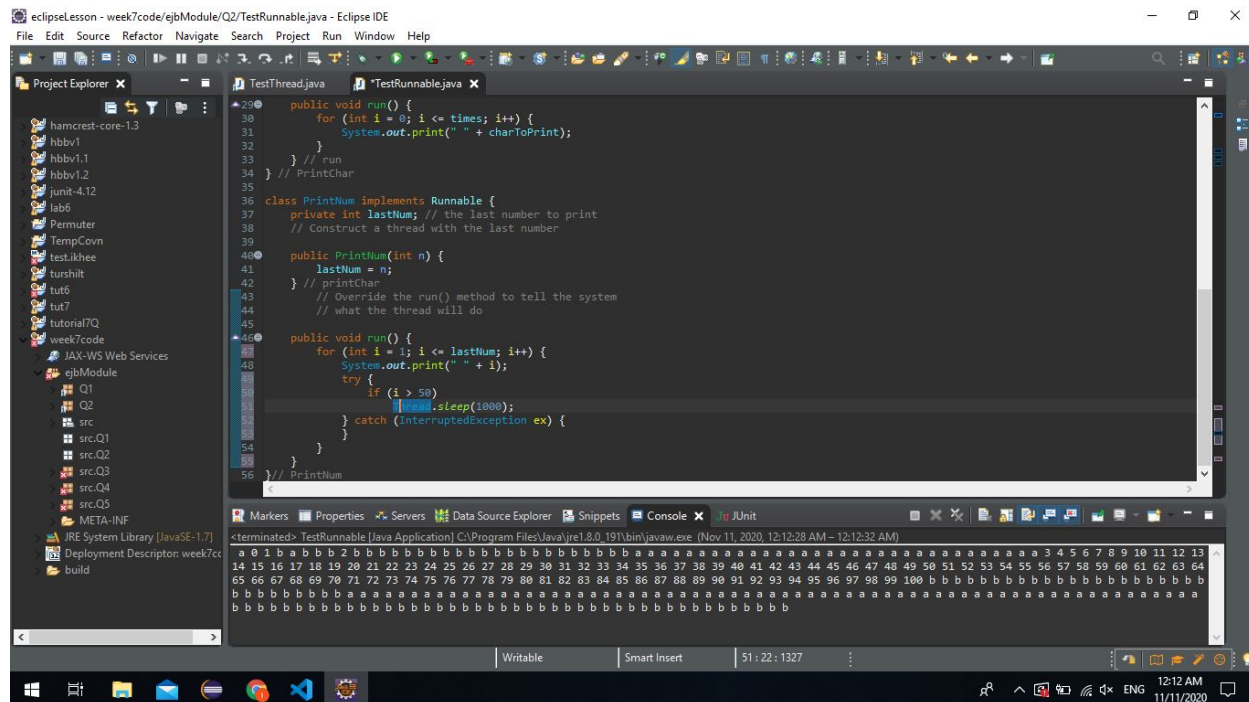
// in specified times

```
class PrintChar extends Thread {
    private char charToPrint; // The character to print
    private int times; // The times to repeat

    /**
    * Construct a thread with specified character and number of times to print the
    * character
    */
    public PrintChar(char c, int t) {
        charToPrint = c;
        times = t;
    }

    public void run() {
        for (int i = 0; i < times; i++)
            System.out.print(charToPrint);
    }
}
```

Q3: TestRunnableSleep.java-д PrintNum классыг өөрчлөн дараагийн хэвлэх тоо нь 50-аас их бол тус thread-ийг 1 секунд унтдаг болго.



Q4: Q1, Q2, Q3 дахь программуудыг ажиллуулахад үр дүн нь тодорхойгүй байна. Яагаад вэ?

Учир нь thread-үүдийн ажиллах дарааллыг тодорхойлж өгөх боломжгүй байдаг.

Q5: Runnable интерфэйсийг хэрэглэх нь Thread классыг ашигласнаас ямар давуу талтай вэ?

Thead бол класс. Хэрвээ түүнийг superclass болгон ашиглавал java нь 1 олон удамшлыг үүсгэж чадахгүй.

Runnable програмыг хэрэгжүүлснээр хүүхэд класст нь хамааралтай болно.

Q6: Ямар хоёр төрлийн thread-үүд байдаг вэ? Тэдгээрийг ялгаа юу вэ?

Daemon and non-daemon threads.

Daemon нь service болон background хэсэгт ажиллаж байдаг thread юм.

NON-Daemon энэ нь хэрэглэгчийн програмын харилцах хэсэг дээр ажилладаг бөгөөд нэг гол thread нь бусад thread үүдтэйгээ холбогдон ажилладаг. NON-Daemon thread ажиллахаа больсон үед програм нь ажиллахаа болино.

Q7: Event-dispatch thread дотор яагаад их цаг зарцуулах үйлдэл гүйцэтгэх ёсгүй вэ?

Учир нь хэрэглэгчтэй харилцах хэсэгт нарийн ажиллагаа ихтэй thread-үүдийг туршиж үзэх нь хэрэглэгчид хариу үзүүлэх үйлдэл удаан болгоход хүргэж болно.

Q8: SwingAlphabet гэсэн шинэ классыг SwingWorker классаас удамшуулан үүсгэ. Энэ нь private хандалттай цагаан толгойг хадгалах тэмдэгтэн цуваа төрөлтэй гишүүн өгөгдөлтэй байна. doInBackground функц, done функц, process функцүүдийг дахин программчилна. Мөн сонгосон үсэг тус бүрийг хэвлэдэг ба эцэст нь тэмдэгтэн цувааг буцаадаг байна.

```
class SwingAlphabet extends SwingWorker<String, Character> {
    Private String alpha="abcdefghijklmnopqrstuvwxyz";
```

```

@Override
protected String doInBackground() {

@Override
protected void process(List<Character> chunks) {

@Override
protected void done() {
}
}

```

Q9: doInBackground() функцийг хэрэгжүүл: alpha гэсэн хувьсагчаас санамсаргүйгээр нэг тэмдэгт сонгож авна, түүнийгээ publish хийж GUI-д дэлгэцлэдэг байна, сонгосон тэмдэгтээ тэмдэгтэн цувааны төгсгөлд залгадаг байна, эцэст нь 1 секунд унтдаг байна. Энэ үйлдлүүдийг 10 удаа давтаж санамсаргүйгээр үүсгэсэн шинэ тэмдэгтэн цувааг буцаана.

```

@Override
protected String doInBackground() {
    String useg = "";
    Random r = new Random();
    for (int i = 0; i < 10; i++) {
        char t = alpha.charAt(r.nextInt(24));
        super.publish(new Character(t));
        useg += t;
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    return useg;
}

```

Q10: done() функцийг хэрэгжүүлж боловсруулагдсан тэмдэгтэн цувааг TraceBox-д залгадаг болго

```

@Override
protected void done() {
    try {
        TraceBox.append("... done: str=" + super.get() + "\r\n");
    } catch (InterruptedException | ExecutionException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    AlphaButton.setEnabled(true);
}

```

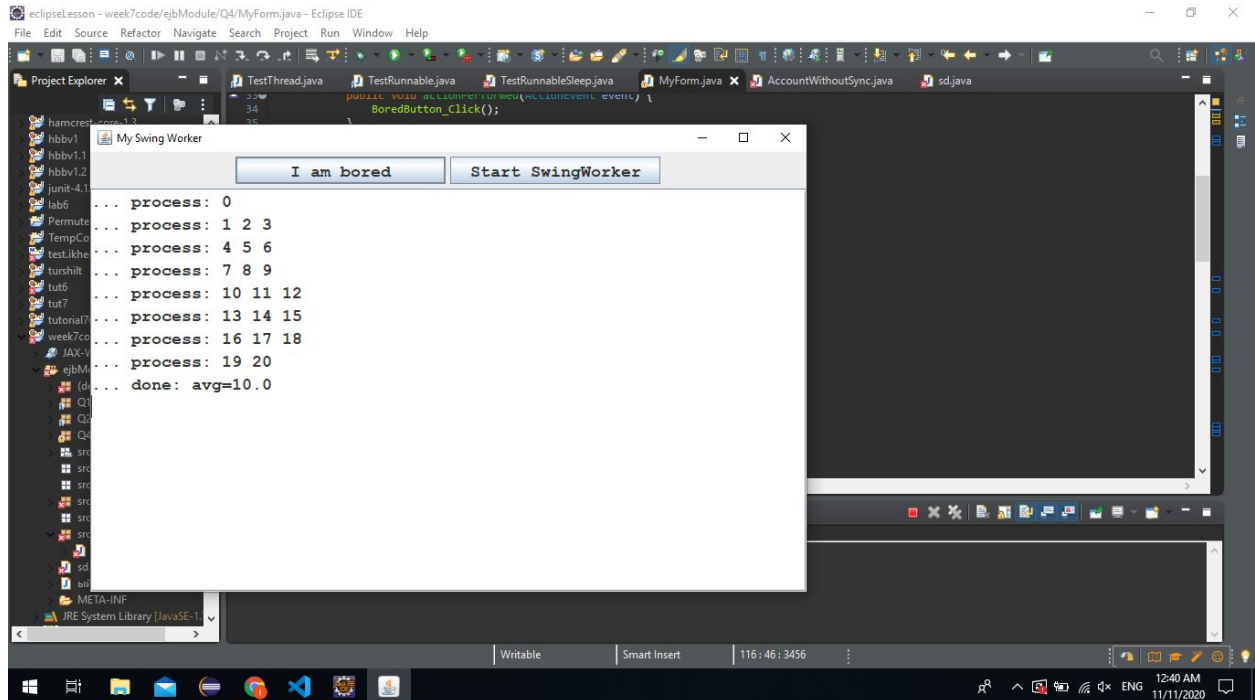
Q11: process() функцийг хэрэгжүүлж сонгогдсон тэмдэгтийг Tracebox-д залгадаг болго.

```

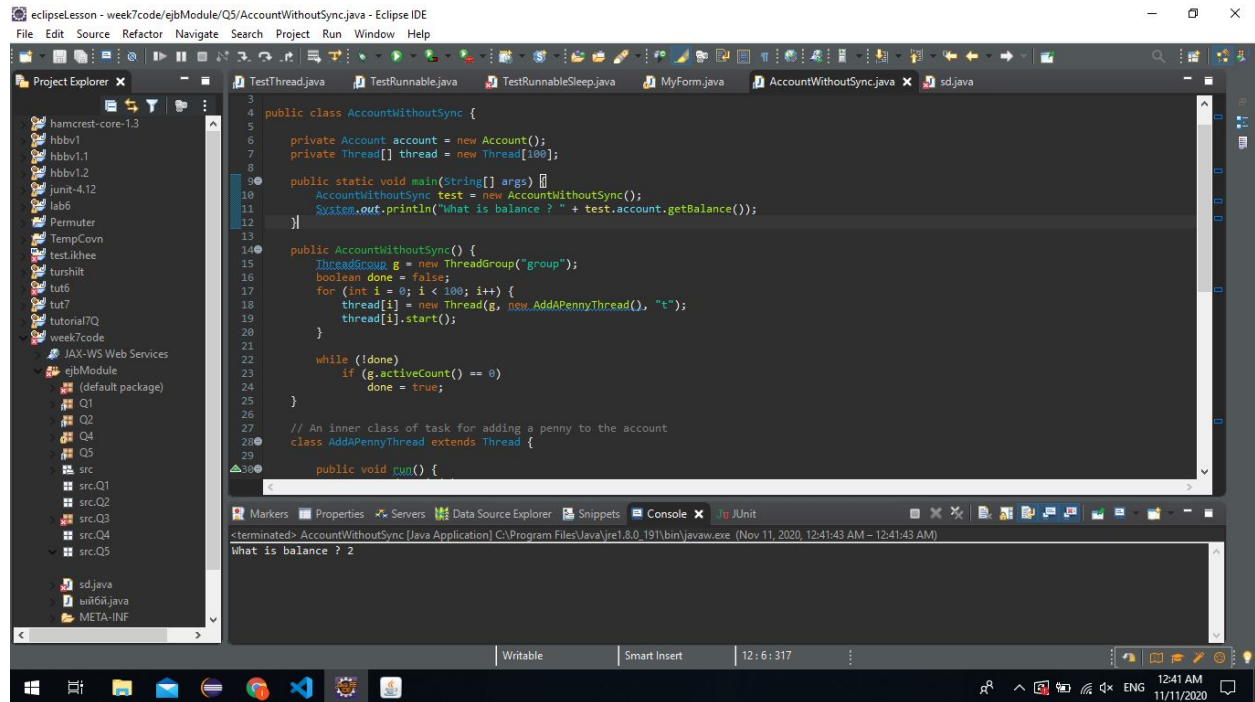
@Override
protected void process(List<Character> chunks) {
    TraceBox.append("... process: ");
    for (Character i : chunks) {
        TraceBox.append(i + " ");
    }
    TraceBox.append("\r\n");
}
}

```

Q12: Эцэст нь, шинэ SwingAlphabet-г эхлүүлэх товчлуурыг нэмж JFrame –н хэмжээг 800,520 болго.



Q13: Q5 пакэйж дэх AccountWithoutSync.java программыг ажигла, хэрэглэгч тус бүр нь 1 цент нэмдэг 100 thread үүсгэхийг хүсч байна. 100 thread ажиллаж дууссаны дараа үлдэгдэл 100 болсон байх ёстой.



```
public class AccountWithoutSync {
```

```
    private Account account = new Account();  
    private Thread[] thread = new Thread[100];
```

```
    public static void main(String[] args) {  
        AccountWithoutSync test = new AccountWithoutSync();  
        System.out.println("What is balance ? " + test.account.getBalance());  
    }
```

```
    public AccountWithoutSync() {  
        ThreadGroup g = new ThreadGroup("group");  
        boolean done = false;  
        for (int i = 0; i < 100; i++) {  
            thread[i] = new Thread(g, new AddAPennyThread(), "t");  
            thread[i].start();  
        }
```

```
        while (!done)  
            if (g.activeCount() == 0)  
                done = true;  
    }
```

```
    // An inner class of task for adding a penny to the account  
    class AddAPennyThread extends Thread {
```

```

        public void run() {
            account.deposit(1);
        }
    }

    // An inner class for account
    class Account {

        private int balance = 0;

        public int getBalance() {
            return balance;
        }

        public void deposit(int amount) {

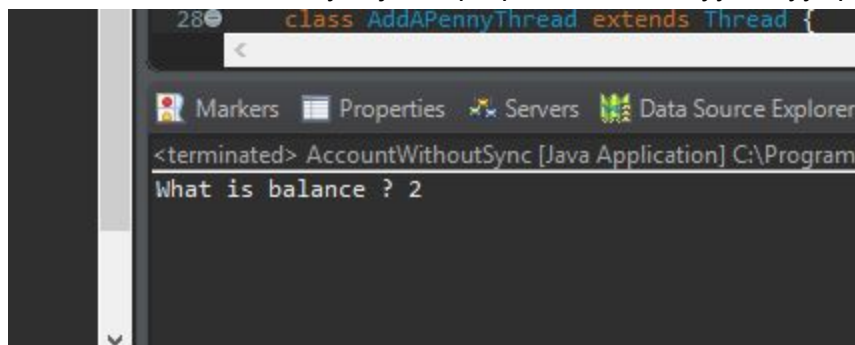
            int newBalance = balance + amount;

            // This delay is deliberately added to magnify the
            // data-corruption problem and make it easy to see.
            try {
                Thread.sleep(5);
            } catch (InterruptedException ex) {
                // do nothing
            }

            balance = newBalance;
        }
    }
}

```

Q13-a: AccountWithoutSync.java программыг ажиллуулна уу, үлдэгдэл хэд байна вэ?



Q13-b: Асуудал юунд байна вэ?

Олон thread нэг object дундаа ашиглаж байгаа нь гол асуудал болж байна.

Q13-c: Үүнийг хэрхэн засах вэ?

Public synchronized void deposit(int amount)

```
43
44 public synchronized void deposit(int amount) {
45
46     int newBalance = balance + amount;
47
48     // This delay is deliberately added to magnify the
49     // data-corruption problem and make it easy to see.
50     try {
51         Thread.sleep(5);
52     } catch (InterruptedException ex) {
53         // do nothing
54     }
55
56     balance = newBalance;
57 }
58
59 }
```

Markers Properties Servers Data Source Explorer Snippets Cor

<terminated> AccountWithoutSync [Java Application] C:\Program Files\Java\jre1.8.0_191\

What is balance ? 100