18

```
public class Test{

public static void main(String[]args) {

// your implementation goes here

Counter c1 = new Counter(5, 4);


new Thread(){

public void run() {

c1.countRange();

}

}.start();


new Thread(){

public void run() {

c1.countRange();

}

}.start();


}

}
```

17

```
public class Test implements Runnable {

public void run() {

try {

System.out.println("Executing run() method");
} catch (Exception e) {

System.out.println(e);

}

}

public static void main(String[] args) {

Thread.currentThread().setPriority(8);

System.out.println("main thread priority is " + Thread.currentThread().getPriority());

Thread t = new Thread(new Test());

System.out.println("myTest thread priority is " + t.getPriority());

}

}
```

19

```
public class Test extends JFrame {
   JButton myButton = new JButton("Test button");
   public static String generateLabel(){
       List<String> labels = new ArrayList<String>();
       labels.add("Test button");
       labels.add("Press me");
       labels.add("Do you feel lucky?");
       labels.add("Try me");
       labels.add("Test me");

       Random random = new Random();
       int randomInteger = random.nextInt(labels.size());
       return labels.get(randomInteger);
   }

   public Test(){
       setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       setTitle("Test app");
       setSize(300,200);
```

```
        setLocation(10,200);

        JPanel content = new JPanel();
       // JButton myButton = new JButton("Test button");
        content.add(myButton);
        setContentPane(content);

        // your code for handling button click event goes here
        myButton.addActionListener(new CustomListener());

    }
private class CustomListener implements ActionListener{
   @Override
   public void actionPerformed(ActionEvent e) {
        myButton.setText(Test.generateLabel());
    }
}
   public static void main(String[]args) {
        JFrame myApp = new Test();
        myApp.show();
    }
}
```

20

```
public class Test {
   private static <K, V, D> void printObject(K keyObject, V valueObject, D descriptionObject) {
         System.out.print(keyObject+": "+valueObject+", "+descriptionObject);
   }

   public static void main(String[] args) {
         printObject("StringKey",101,"new third argument");
         System.out.println();

         printObject(10, "this is my value", 5);
         System.out.println();

         printObject("StringKey", 10.5, "new argument");
         System.out.println();
```

```
        printObject(12, 1.2, "really new argument");
        System.out.println();
  }
}
```

22

```
public class Test {

public class Test {

public static void main(String args[]){

// your code goes in here; create a new instance of Dog //class, call it "dog1"

try{

Dog dog1 = Dog.class.newInstance();

dog1.setName("Ki");

dog1.setWeight(100.2);

}catch(InstantiationException | IllegalAccessException e){

System.out.println(e);

}

}

}
```

21

```
Class employeeClass = Class.forName("Employee");
```