

BCP-811

**BCP-811**

**Developing Java Applications for the BlackBerry  
Platform**

Version 3.5

## BCP-811

## Topic 1, Volume A

## QUESTION NO: 1

Which block of code will ensure that a network connection has been closed? (Choose one.)

A.

```
HttpURLConnection conn = null;
OutputStream out = null;
InputStream in = null;
String URL = "http://www.blackberry.com";
try {
    conn = (HttpURLConnection)Connector.open(URL);
    out = conn.getOutputStream();
    in = conn.getInputStream();

} catch( IOException e ) {

} finally {
    if (in != null) {
        try {
            in.close();
        } catch (IOException e) {
        }
    }
    if (out != null) {
        try {
            out.close();
        } catch (IOException e) {
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (IOException e) {
        }
    }
}
```

## BCP-811

C B.

```
HttpConnection conn = null;
OutputStream out = null;
InputStream in = null;
String URL = "http://www.blackberry.com";
try {
    conn = (HttpConnection)Connector.open(URL);
    out = conn.openOutputStream();
    in = conn.openInputStream();

} catch( IOException e ) {

}
in.close();
out.close();
conn.close();
```

C C.

```
HttpConnection conn = null;
OutputStream out = null;
InputStream in = null;
String URL = "http://www.blackberry.com";
try {
    conn = (HttpConnection)Connector.open(URL);
    out = conn.openOutputStream();
    in = conn.openInputStream();

} catch( IOException e ) {

}

if (in != null) {
    try {
        in.close();
    } catch (IOException e) {
    }
}
```

## BCP-811

---

C D.

```
HttpConnection conn = null;
OutputStream out = null;
InputStream in = null;
String URL = "http://www.blackberry.com";
try {
    conn = (HttpConnection)Connector.open(URL);
    out = conn.openOutputStream();
    in = conn.openInputStream();
} catch( IOException e ) {

} finally {
    if (in != null) {
        in.close();
    }
    if (out != null) {
        out.close();
    }
    if (conn != null) {
        conn.close();
    }
}
```

## BCP-811

C E.

```
HttpConnection conn = null;
OutputStream out = null;
InputStream in = null;
String URL = "http://www.blackberry.com";
try {
    conn = (HttpURLConnection)Connector.open(URL);
    out = conn.openOutputStream();
    in = conn.openInputStream();

} catch( IOException e ) {
    if (in != null) {
        try {
            in.close();
        } catch (IOException e) {
        }
    }
    if (out != null) {
        try {
            out.close();
        } catch (IOException e) {
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (IOException e) {
        }
    }
}

if (in != null) {
    try {
        in.close();
    } catch (IOException e) {
    }
}

if (out != null) {
    try {
        out.close();
    } catch (IOException e) {
    }
}

if (conn != null) {
    try {
        conn.close();
    } catch (IOException e) {
    }
}

if (in != null) {
    try {
        in.close();
    } catch (IOException e) {
    }
}

if (out != null) {
    try {
        out.close();
    } catch (IOException e) {
    }
}

if (conn != null) {
    try {
        conn.close();
    } catch (IOException e) {
    }
}
}
```

## BCP-811

- A. Exhibit A
- B. Exhibit B
- C. Exhibit C
- D. Exhibit D
- E. Exhibit E

**Answer: A**

**QUESTION NO: 2**

Consider the code below:

```
int width = Math.abs( ... );  
...  
int midpoint = width / 2;
```

What is a more efficient way of calculating the midpoint? (Choose one.)

- A. `int midpoint = (int) ((double) width) / 2.0;`
- B. `int midpoint = Fixed32.div(width, 2);`
- C. `int midpoint = (int) (width * 0.5f);`
- D. `int midpoint = width >> 1;`
- E. `int midpoint = width >> 2;`

**Answer: D**

**QUESTION NO: 3**

Which two of the following operations should be avoided on the main event thread? (Choose two.)

- A. Pushing a screen onto the display stack
- B. Displaying a Dialog screen

## BCP-811

- C. Performing network communication
- D. Calling the sleep method
- E. Capturing of keyboard input

**Answer:** C,D

**QUESTION NO: 4**

A BlackBerry device application collects information about several hundred books. The application needs to sort books by title. Which combination of data structures should be used to manage the objects? (Choose one.)

- A. `net.rim.device.api.util.SimpleSortingVector` and `net.rim.device.api.util.StringComparator`
- B. `java.util.Hashtable` and `net.rim.device.api.util.StringComparator`
- C. `net.rim.device.api.util.StringRepository` and `net.rim.device.api.util.StringPattern`
- D. `java.util.TreeSet` and `net.rim.device.api.util.StringComparator`
- E. `java.util.Stack` and `net.rim.device.api.util.StringComparator`

**Answer:** A

**QUESTION NO: 5**

An application receives notification that a new data object is being saved in the `RuntimeStore`. It may take up to 60 seconds to save this data. Which method should the application use to obtain the object from the `RuntimeStore`? (Choose one.)

- A. `get( long objectUID )`
- B. `getInstance()`

## BCP-811

- C. fetch( Class objectClass, long timeout )
- D. waitFor( long objectUID )
- E. put( long objectUID, Object anObject)

**Answer: D**

**QUESTION NO: 6**

The following application is set to auto-run at startup:

```
public final class TestApp extends UiApplication
{

    public static void main(String[] args)
    {
        UiApplication theApp = new TestApp();
        theApp.enterEventDispatcher();
    }

    public TestApp()
    {
        MainScreen mainScreen = new MainScreen();

        MenuItem menu = new MenuItem("Do Something",
            1, 1)
        {
            public void run()
            {
                MyThread myTh = new MyThread();
                myTh.start();
            }
        };
        mainScreen.addMenuItem(menu);
        pushScreen(mainScreen);
    }
}
```



## BCP-811

Which approach would most reliably solve the problem this application will encounter? (Choose one.)

- A. It should call `Thread.sleep()` for ten seconds before pushing the screen to verify that the BlackBerry device has completed its boot up process
- B. Threads should be moved to their own standalone class so that they do not cause a security exception
- C. The int values in the `MenuItem` constructor should equal the `MenuItem` length to leave enough room for the string
- D. It should use the `ApplicationManager.inStartup()` before pushing the screen to verify that the BlackBerry device has completed its boot up process
- E. It should use the `ApplicationManager.inStartup()` before pushing the screen to verify that the BlackBerry device has completed its boot up process
- F. It should use the `ApplicationManager.inStartup()` before pushing the screen to verify that the BlackBerry device has completed its boot up process
- G. The `makeMenu` method of `MainScreen` must be overridden to use a menu

**Answer:** D,E,F

**QUESTION NO: 7**

A developer is designing a BlackBerry device application that provides summaries of large amounts of data. The summaries require extensive computations. Which design approach will meet these requirements? (Choose one.)

- A. Store the full set of data locally on removable media using a compressed format
- B. Supplement memory available on the BlackBerry device and compute the results locally
- C. Use URL encodings to expedite the transfer of data between a remote server and the BlackBerry device
- D. Compute the summary on a remote server and access the results through a BlackBerry Enterprise Server connection

## BCP-811

E. Use local storage and the floating point processor on the BlackBerry device to speed up computations

**Answer: B**

**QUESTION NO: 8**

A customer needs an application that will store temporary data and expose it to other BlackBerry device applications. Which API will meet this requirement? (Choose one.)

- A. RuntimeStore
- B. PersistentStore
- C. Location Based Services
- D. Global Events
- E. Connector

**Answer: A**

**QUESTION NO: 9**

An application requires the creation of a custom field that can display animation. Which class and method combination should be used to accomplish this task? (Choose one.)

- A. Field.paint(Graphics graphics)
- B. MainScreen.paint(Graphics graphics)
- C. Field.subpaint(Graphics graphics)
- D. Screen.draw(Graphics graphics)
- E. GameCanvas.paint(Graphics graphics)