1. What is printed when printSomething() is called as shown below:

```java
public static void main(String... args) {
    printSomething(10, 10, 11);
}

public static void printSomething(int x, int y, int z) {
    if (x++ > y) {
        if (y < z) {
            System.out.println("Output 1");
        } else {
            System.out.println("Output 2");
        }
    } else {
        if (++y == z) {
            System.out.println("Output 3");
        } else {
            System.out.println("Output 4");
        }
    }
}
```

Answer: Output 3

2. The code below produces indeterminate output each time it runs. Explain why and suggest a way of fixing the program so that it produces consistent output.

```java
int a, b;
for (int i = 0; i < 10; i++) {
    if (a == 0) { // a contains shared data
        a = 1;
        b = a;
    }
}
```

a and b contain shared data between two threads. Between the check (a==0) and the assignment a=1, a thread could be paused. The other thread could then pass the same check, then read the same value of
b and increment b concurrently with the other thread, losing an increment.

Solution: Remove all but 'b++' from incrementB and enclose it in synchronized(this) {  b++; }. Making incrementB synchronized also works.

```java
public class BasicThreadsafetyExample {
    volatile int a = 0;
    int b = 0;

    public void incrementB() {
        if (a == 0) {
            a = 1;
            b++;
            a = 0;
        } else {
            try {
                Thread.sleep(250);
            } catch (InterruptedException e) {} 
            incrementB();
        }
    }

    public static void main(String... args) {
        new BasicThreadsafetyExample().test();
    }

    private void test() {
        Runnable runnable1 = new Runnable() {
            public void run() {
                for (int i = 1; i <= 100_000; i++) {
                    incrementB();
                }
            }
        };
    }
}
```
Thread thread1 = new Thread(runnable1);
thread1.start();
Thread thread2 = new Thread(runnable1);
thread2.start();

try {
    thread1.join(); // wait for thread1 to complete
    thread2.join(); // wait for thread2 to complete
} catch (InterruptedException e) {
}

System.out.println("Counter = " + counter);

3. You want to start up a social media website and decide to write a Java server application to support it.

The SocialMediaApp class keeps track of the system state, including the list of logged-in users.

The User class keeps track of user information, the user's Socket connection, and the user's list of friends. When one of a User's friends logs in or out, the User class's friendOnline() or friendOffline() method respectively, is called.

Similarly, when a user logs in or out, the SocialMediaApp calls the login() or logOff() method, respectively. Implement login() and logoff() below.

class SocialMediaApp {
    private static List<User> users = new ArrayList<>();

    // methods to add/remove users when they log in and out
    public static add(User) { ... }
    public static remove(User) { ... }
}
public static List<User> getOnlineUsers() {
    return users;
}

class User {
    int id;
    String name;
    String phone;
    String address;
    String city;
    String state;
    String zipCode;
    Socket connection;

    List<Friend> friends = new ArrayList<>();

    sendMessage( String message ) {
        if (connection != null) {
            connection.writeMessageToUser( message );
        } else
            throw new IllegalStateException("Cant send message, User is not logged in");
    }

    friendOnline(Friend friend) {
        friends.add(friend);
        sendMessage("Your friend: " + friend.name + " just logged in! ");
    }

    friendOffline(Friend friend) {
        sendMessage("Your friend: " + friend.name + " just logged out! ");
        friends.remove(friend);
    }

    // Write the code to notify other logged-in users that their friend has logged in.
login() {
    for (User user : SocialMediaApp.getOnlineUsers()) {
        if (user.isFriend(this))
            user.friendOnline(this);
    }
}

// Write the code to notify other logged-in users that their friend has logged out.
logOff() {
    for (User user : SocialMediaApp.getOnlineUsers()) {
        if (user.isFriend(this))
            user.friendOffline(this);
    }
}

boolean isFriend(Friend f) {
    return friends.contains(f);
}

4. Explain how a working Swing user interface is constructed from JComponents and can pass data collected from those components to an instance of an application class.

a. Select and configure the JComponents necessary to meet user interaction requirements.
b. Create an EventHandler for each component for which you wish to detect an event. In the event handler, use the event object and or get the JComponent state and pass it to your application object(s) to implement your app.

5. What are two advantages of using a List<String> over an array String[]?

a. A list will automatically resize if you add to a list that's at capacity.
b. The list's implementation can change in one line of code, unlike the array's.
   c. List's have built-in methods to do things like add an element at some
position and shift the other items in the list, unlike an array where you'd need to implement that yourself.

6. What are two differences between a java.util.List and a java.util.Set? How does a java.util.Map differ from either of them?

a. Lists may contain duplicates, unlike Sets.
b. Lists have ordering based on position while sets have ordering based on a Comparator (compareTo method).

7. Show the code that connects the two streams below so that any data that arrives on a Socket is written to a file:
( the read() method of a stream will return -1 if the end of stream has been reached ).

```java
private void writeSocketContentToFile(Socket s, FileWriter f) {
    try {
        BufferedReader br = new BufferedReader(s.getInputStream());
        String line = br.readLine(); // this gets the next line from socket

        // If line is null, end of stream has been reached
        // Add the code to read from the socket until end of stream
        // and write the contents to the file
        while (line != null) {
            f.write(line);
            line = br.readLine();
        }

        // close the file when done writing to it
        f.close();
    } catch (IOException e) {
        System.out.println("IO Exception: " + e);
    }
}
```
8. The table below represents a set of inventory:

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity on Hand</th>
<th>Price each</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>$10.25</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>$2.25</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>$125.95</td>
</tr>
</tbody>
</table>

Create a class that represents an Inventory Item.

```java
class InventoryItem {
    int number;
    int quantity;
    double price;
}
```

Create a Report class with a method that accepts as input any set of inventory items and returns the total value of the inventory.

```java
class Report {
    public double computeTotalValue(List<Item> items) {
        double value = 0.0d;
        for (Item item : items) {
            value += item.quantity * item.price;
        }
        return value;
    }
}
```

9. The compiler generates an error that there's an unhandled FileNotFoundException in this method:
public void tryToCreateFile(String fileName) {
    FileOutputStream fos = new FileOutputStream(fileName);
}

What are two different ways of handling the exception?

a. throws FileNotFoundException from method signature line
b. try { } catch (FileNotFoundException fnfe) { }

10. The compiler reports an error in the code below:  
"local variable x may not have been initialized".

    public void definiteProblem(int y) {
        int x;
        if (y < 0) {
            x = 1;
        }
        else if (y >= 0) {
            x = 2;
        }

        System.out.println("X = " + x);
    }

What are two ways to correct the error without changing the intent of the code?

a. int x = 0;
b. else if -> else
11. What is a Swing layout manager used for?

A layout manager is used to position Swing components (and panels/frames) within a panel or frame. The layout manager uses a Policy to decide where to place the components with respect to each other and how to size them when certain events occur.

12. What are the two main advantages of using threads?

a. To block waiting for an event to occur without blocking the user's thread so a user will not be kept waiting for I/O or other event.

b. To partition work into different fragments to be completed in parallel, assuming multiple processors are available to do the work, thereby speeding up the computations.

13. Suppose you are writing classes to support a video game which follows these rules:

a. A 700x500 pixel screen contains three types of objects: asteroids, a player's spaceship, and an alien spaceship.

b. There is one player who controls the spaceship. The ship is equipped with a weapon that can emit a ray to sever asteroids in front of it.

c. The ray propagates rapidly (visually on the display) from the front of the spaceship and continues in a line until it either hits another object or goes off the screen.

d. If the ray hits an asteroid, the asteroid severs into several smaller asteroids each moving off in a random direction.

e. If a small asteroid is hit by the ray, it will disappear from the screen.

f. The player receives points for hitting any asteroid with the ray.

g. If the ray hits an alien spacecraft, it will move off the board and the player's score increases.

h. The player's spacecraft can rotate in either direction (when the user...
presses one of two different keyboard keys) or apply thrust to move forward when the user presses a third key.
i. If the ship moves off the board, it reappears on the opposite side. If the thrust key is pressed, the ship will accelerate then gradually slow to a stop.
j. The score should appear somewhere on the screen outside the 700x500 area.
k. If the ship collides with an asteroid, the game ends.

What primary classes would you create to implement this game?
List the class names and a description of each's purpose.

- Asteroid
- Spaceship
- AlienShip
- Ray
- ScoreKeeper

What methods would add to implement the rules listed above? List the method names and a description of each's purpose.

Game
  run()
  checkCollision()
  computeNextPositions()
  addToScore()
  updateScoreDisplay()

GameDisplay
  manageGameTime()
  update()

Ray:
  moveRay()
  disappear()
Asteroid:
    rayHit()
    split()
    disappear()

Spaceship:
    fireRay()
    rotate()
    forwardThrust()

AlienShip()
    rayHit()
    moveOffScreen()
    moveAroundScreen()

14. What does this program print?

X = 10
a = 12
X = 10
b = 5

public abstract class WhatDoesItPrint {
    int x = 10;
    String a = "hello";

    protected abstract void andThis();

    public static void main(String[] args) {
        WhatDoesItPrint w = new Another();
        w.tryThis(10);
        w.andThis();
    }
}
protected void tryThis(int b) {
    int a = 12;
    System.out.println("X = " + x);
    if (x == b) {
        System.out.println("a = " + a);
    } else {
        System.out.println("b = " + b);
    }
}

class Another extends WhatDoesItPrint {
    int x = 5;
    protected void andThis() {
        this.tryThis(x);
    }
}