1) Examine class below for null pointer/ null value aspect and mark lines or code sections you think it has risk.

```
public class Sample {

    public List<Character> doSomething(List<Character> charList) {

        List<Character> returnList = null;
        for (Character c : charList) {
            if (Character.isDigit(c)) {
                if (returnList == null)
                    returnList = new LinkedList<>();
                returnList.add(c);
            }
        }
        return returnList;

    }
}
```

2) What will method1 will print to console in the class below?

```
public class TryCatch {

    public void method1() {

        try {
          badMethod();
          System.out.print("A");
          }
          catch (RuntimeException e) {
          System.out.print("B");
          }
          catch (Exception e) {
                  System.out.print("C");
          }
          finally {
                  System.out.print("D");
          }

          System.out.print("E");
    }

    private void badMethod(){
        throw new RuntimeException();
    }
}
```

3) What will method2 will print to console in the class below?

```
public class TryCatch {

    public void method2() {
        try {
        badMethod();
        System.out.print("A");
          }
     catch (Exception e) {
        System.out.print("B");
          }
          finally {
              System.out.print("C");
          }

          System.out.print("D");
  }

  private void badMethod(){
    throw new Error();
  }
}
```

4) What is wrong with the code below?

```
public class TryCatch {

  public Integer somethingWrongWithThisMethod() {
    try {
      badMethod1();
      return 1;
    }
    catch (RuntimeException e) {
      System.out.println("A Runtime Exception Occurred");
      return 2;
    }
    catch (Exception e) {
      System.out.println("An Exception Occurred");
      return 3;
    }
    finally {
      return 4;
    }
  }

  private void badMethod(){
    throw new Exception();
  }
}
```

5) What would be the output of print(1000) ?

```
public void print(int n) {
        if (n > 4000)
            return;
        System.out.printf("%d ", n);
        print(2*n);
        System.out.printf("%d ", n);
}
```

6) What would be the output of func(11) ?

```
public int func(int n) {
        if (n <= 1)
            return 1;
        if (n % 2 == 0)
            return func(n / 2);
        return func(n / 2) + func(n / 2 + 1);
}
```

7) The class below breaks the design principle "Open for extension, closed for modification". How would you fix the design so that it sticks with mentioned principle ?

```
public class Player {
 private Weapon weapon;
 private String name;

 public Player(String name, Weapon weapon) {
        this.name = name;
        this.weapon = weapon;
 }

 public void setWeapon(Weapon weapon) {
        this.weapon = weapon;
 }

 public void action() {
        if (this.weapon.type == "knife") {
                System.out.println("Perform knife attack");
        } else if (this.weapon.type == "revolver") {
                System.out.println("Perform revolver attack");
        } else if (this.weapon.type == "Plasma Gun") {
                System.out.println("Perform plasma gun attack");
        }
 }
}
```

8) A bakery is using a software to fine tune how many pies to bake each day. It doesn't want to run out of pies to sell, but it also doesn't want too many pies left unsold at the end of the day. Now, you are asked to test a method
int getNumberOfPiesToBake(int leftOverFromYesterday).
The method takes the number of pies leftover at the end of the day yesterday, returns the change in the number of pies we should bake today
The algorithm for the change in the number of pies to bake is:
If 0 pies were left over, increase the number of pies we bake by 40
If 1-20 pies were left over, don't change how many pies we bake
If more than 20 pies were left over, reduce the number of pies we bake today by: the amount of pies left over yesterday, minus 20
For which values you would plan to test this method. What would your input values be?

9) Please implement a java program which prints to console the output below. You will have 10 minutes for it.

```
*                *

**              **

***            ***

****          ****

**********
```

10) Write a method which calculates angle between hour and minute hands in an analog clock.
Input for hour is between 1 – 12
Input for minute is between 0 – 59
You will have 20 minutes for this question.

```
public double calculateAngle(int hour, int minute){
        //implement method
}
```