CS: 114 **OOPDA** **Spring, 2025**

Project5 (100 points) **Due 4/9/2025**

Do not send the entire project. All I want are the files ending in .java, the UML diagrams with class relationships, and any audio file from problem P3. Please make sure your name is included at the top of the source code. Use a comment statement. Upload a zip file containing all of your files to canvas. There should be one driver class called Project5_Driver.java.

The grading breakdown for this project is as follows:

10% Readability – Is the program easy to read and understand (indentation, documentation, good use of white space, good output format, user prompts)

10% Java – Does the program make good use of the Java constructs (functions, control flow, etc.)

30% Robustness - Does the program compile and run, and not crash or throw exceptions

50% Correctness – Does the program solve the intended problem and work on a variety of reasonable inputs

Create UML diagrams and make sure your code is javadoc compliant.

Use the `Project5_driver.java` file as a template for P1 to P3 below.

Problem1

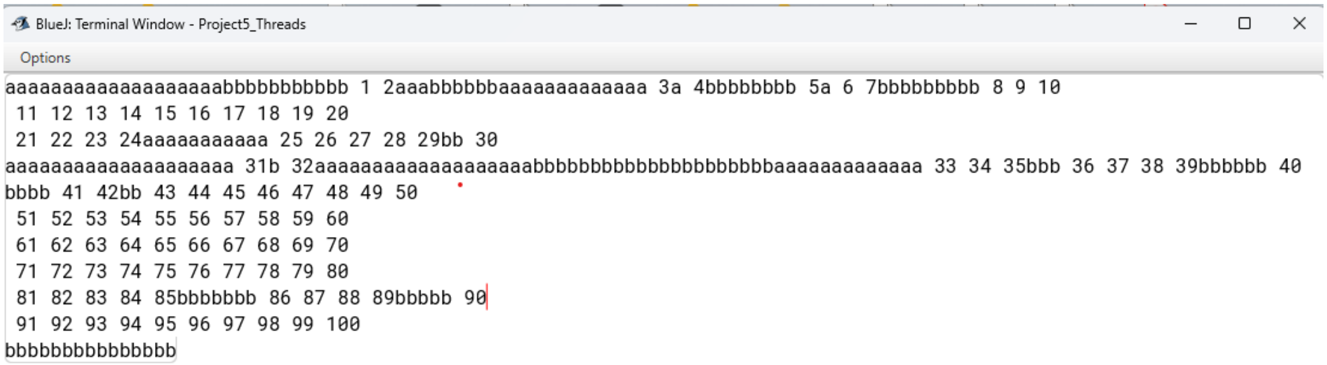
Examine the `P1` code below. Then implement the two classes `PrintChar` and `PrintNum`. Both of these classes should implement a `Runnable` interface and override the `run` method. The `PrintChar` method has two arguments. The first argument is of type `char` and represents the character to print. The second argument is of type `int` and represents the number of times to print the character with a call to `System.out.print` inside a loop.

The `PrintNum` method will print the numbers from 1 to `n,` where `n` is the one argument passed into to `PrintNum`. After every 10 numbers are printed a carriage return will skip to the beginning of the next line.

```
public static void P1()
{
   Thread printA   = new Thread(new PrintChar('a', 100));
   Thread printB   = new Thread(new PrintChar('b', 100));
   Thread print100 = new Thread(new PrintNum(100));

   printA.start();
   printB.start();
   print100.start();
}
```

Output from P1 might look like this



```
aaaaaaaaaaaaaaaaaaabbbbbbbbbbb 1 2aaabbbbbbaaaaaaaaaaaaa 3a 4bbbbbbbb 5a 6 7bbbbbbbbb 8 9 10
 11 12 13 14 15 16 17 18 19 20
 21 22 23 24aaaaaaaaaaa 25 26 27 28 29bb 30
aaaaaaaaaaaaaaaaaaaa 31b 32aaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbaaaaaaaaaaaa 33 34 35bbb 36 37 38 39bbbbbb 40
bbbb 41 42bb 43 44 45 46 47 48 49 50
 51 52 53 54 55 56 57 58 59 60
 61 62 63 64 65 66 67 68 69 70
 71 72 73 74 75 76 77 78 79 80
 81 82 83 84 85bbbbbbb 86 87 88 89bbbbb 90
 91 92 93 94 95 96 97 98 99 100
bbbbbbbbbbbbbbb
```

Problem2

Move the code from `ex3` in the lect9 sample code to `P2` in the `Project5_driver`. Also move over the classes `SavingsAccount`, `WithdrawRunnable`, and `DepositRunnable`, and rename them to `SavingsAccountP5, WithdrawRunnableP5`, and `DepositRunnableP5`. Modify the code to take care of any racing conditions and deadlock conditions per the recommendations that were suggested in lect9.

Problem3

For `P3` implement a music jukebox utilizing the audio tools that were discussed in lect9. Your program should initially load in 5 or more songs and create 5 Runnable objects. Come up with a way to present what songs are available to play. Program a way for the user to select one of the songs and then play the song. The user should be able to stop a song at any time and pick a new song. The user can also listen to a song until it is finished playing. At the point the user should be able to pick a new song to play or exit the program. You are required to use the `audioPlayer` class.