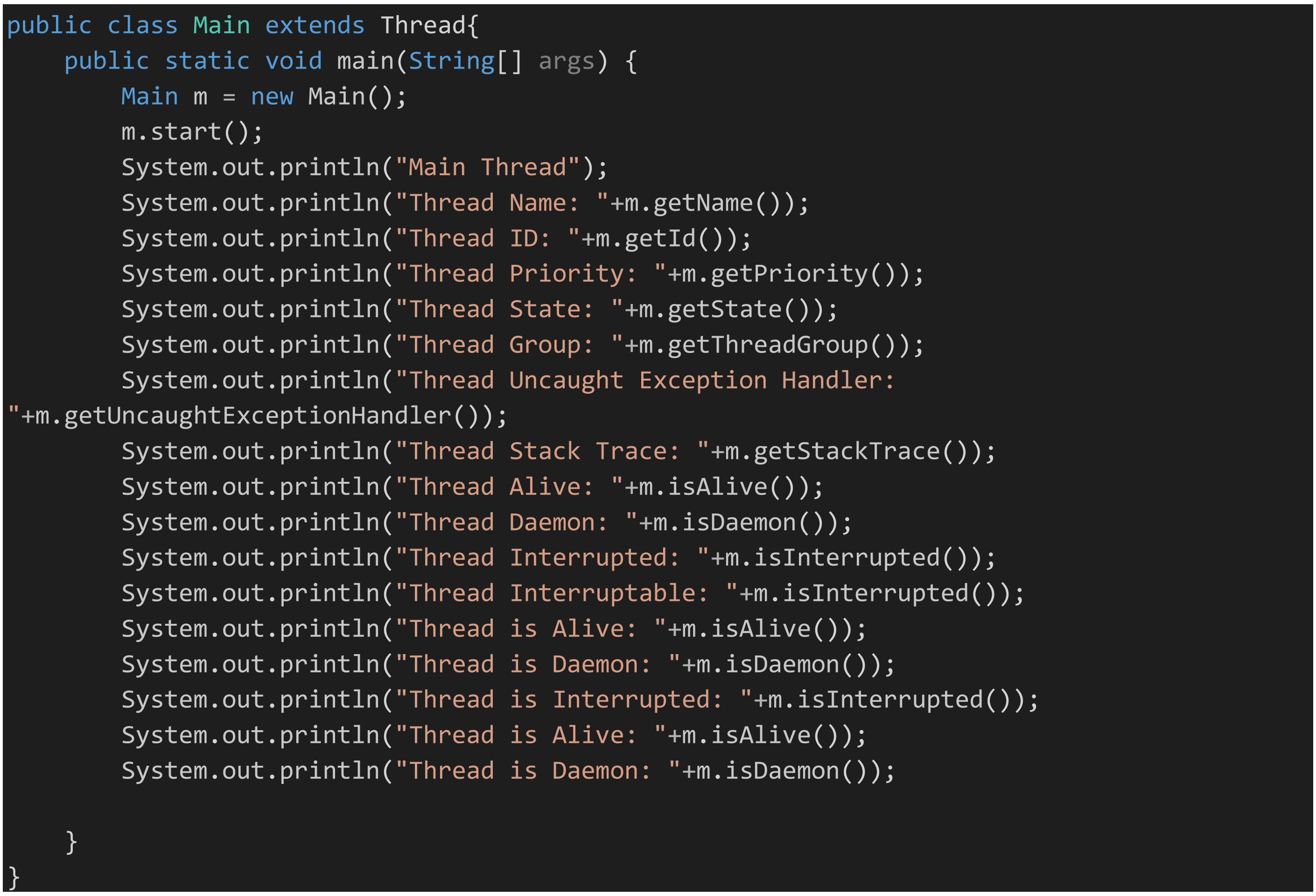
Assignment 3:

User Assignment-4

Problem Statement:

1. Write a java program to create a thread by using Thread class and also by using the Runnable interface and display the details of thread like, thread name, id, priority, its phase and other details.
2. Write a java program to create three different threads, with first thread displaying numbers from 101 to 200, second from 201 to 300 and third one from 301 to 400 and verify that all the threads are running simultaneously or not.

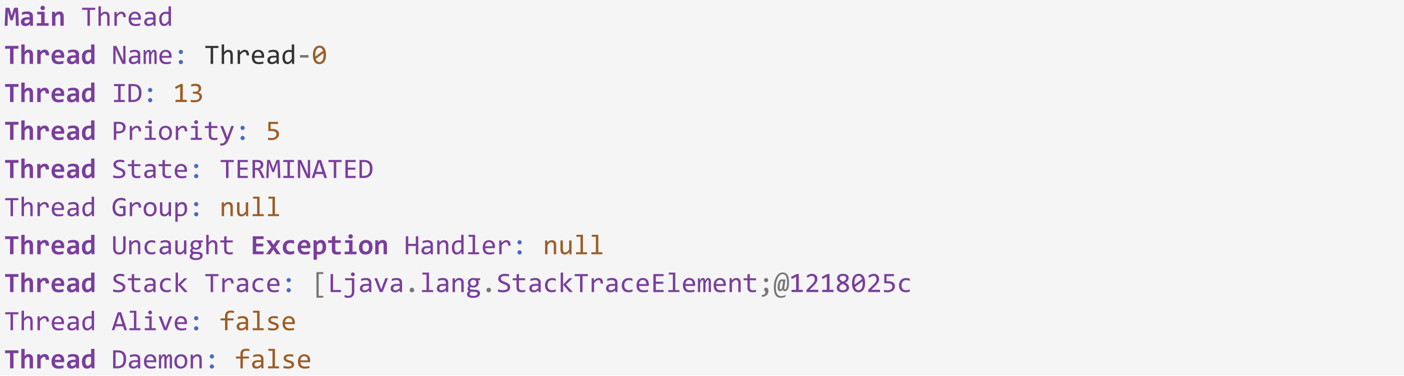
Source Code:



```java
public class Main extends Thread{
    public static void main(String[] args) {
        Main m = new Main();
        m.start();
        System.out.println("Main Thread");
        System.out.println("Thread Name: "+m.getName());
        System.out.println("Thread ID: "+m.getId());
        System.out.println("Thread Priority: "+m.getPriority());
        System.out.println("Thread State: "+m.getState());
        System.out.println("Thread Group: "+m.getThreadGroup());
        System.out.println("Thread Uncaught Exception Handler:
"+m.getUncaughtExceptionHandler());
        System.out.println("Thread Stack Trace: "+m.getStackTrace());
        System.out.println("Thread Alive: "+m.isAlive());
        System.out.println("Thread Daemon: "+m.isDaemon());
        System.out.println("Thread Interrupted: "+m.isInterrupted());
        System.out.println("Thread Interruptable: "+m.isInterrupted());
        System.out.println("Thread is Alive: "+m.isAlive());
        System.out.println("Thread is Daemon: "+m.isDaemon());
        System.out.println("Thread is Interrupted: "+m.isInterrupted());
        System.out.println("Thread is Alive: "+m.isAlive());
        System.out.println("Thread is Daemon: "+m.isDaemon());

    }
}
```

Output:



```
Main Thread
Thread Name: Thread-0
Thread ID: 13
Thread Priority: 5
Thread State: TERMINATED
Thread Group: null
Thread Uncaught Exception Handler: null
Thread Stack Trace: [Ljava.lang.StackTraceElement;@1218025c
Thread Alive: false
Thread Daemon: false
Thread Interrupted: false
Thread Interruptable: false
Thread is Alive: false
Thread is Daemon: false
Thread is Interrupted: false
Thread is Alive: false
Thread is Daemon: false
```

```java
public class Main implements Runnable{
    @Override
    public void run() {
        System.out.println("Thread name: "+Thread.currentThread().getName());
        System.out.println("Thread id: "+Thread.currentThread().getId());
        System.out.println("Thread priority: "+Thread.currentThread().getPriority());
        System.out.println("Thread state: "+Thread.currentThread().getState());
        System.out.println("Thread group: "+Thread.currentThread().getThreadGroup());
        System.out.println("Thread context class loader: 
"+Thread.currentThread().getContextClassLoader());
        System.out.println("Thread stack trace: "+Thread.currentThread().getStackTrace());
        System.out.println("Thread interrupted status: 
"+Thread.currentThread().isInterrupted());
        System.out.println("Thread daemon status: "+Thread.currentThread().isDaemon());
        System.out.println("Thread alive status: "+Thread.currentThread().isAlive());
        System.out.println("Thread interrupted status: 
"+Thread.currentThread().isInterrupted());
    }
    public static void main(String[] args) {
        Thread t1 = new Thread(new Main());
        t1.start();
    }
}
```

Output:

```
Thread name: Thread-0
Thread id: 13
Thread priority: 5
Thread state: RUNNABLE
Thread group: java.lang.ThreadGroup[name=main,maxpri=10]
Thread context class loader: jdk.internal.loader.ClassLoaders$AppClassLoader@73d16e93
Thread stack trace: [Ljava.lang.StackTraceElement;@2ec674c9
Thread interrupted status: false
Thread daemon status: false
Thread alive status: true
Thread interrupted status: false
```

2)

```java
public class Main{
    public static void main(String[] args) {
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                for(int i=101;i<=200;i++){
                    System.out.print(i+" ");
                }
            }
        });
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                for(int i=201;i<=300;i++){
                    System.out.print(i+" ");
                }
            }
        });
        Thread t3 = new Thread(new Runnable() {
            @Override
            public void run() {
                for(int i=301;i<=400;i++){
                    System.out.print(i+" ");
                }
            }
        });
        t1.start();
        t2.start();
        t3.start();
        try{
            t1.join();
            t2.join();
            t3.join();
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("All threads are running simultaneously");
    }
}
```

Output:

```
301 302 303 304 305 201 101 202 306 203 102 204 307 205 103 206 308 207 104 208 309 209
105 210 310 211 106 212 311 213 107 214 312 215 108 216 313 217 109 218 314 219 110 220
315 221 111 222 316 223 112 224 317 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 113 114 115 241 318 242 116 243 319 244 117 245 320 246 118 119 247 321 248
120 249 322 250 251 121 252 323 324 325 326 327 328 329 330 331 332 333 334 253 122 254
335
255 123 256 336 257 124 258 337 259 260 125 261 338 262 126 263 339 264 127 265 340 266
128 267 341 268 129 269 342 270 130 271 343 272 131 273 344 274 132 275 345 276 133 277
```

```
346 278 134 279 347 280 135 281 348 282 136 283 349 284 137 285 350 286 138 287 351 288
139 289 352 290 140 291 353 292 141 293 354 294 142 295 355 296 143 297 356 298 144 299
357 300 145 358 146 359 147 148 360 149 361 150 362 363 151 364 152 365 153 154 366 155
367
156 368 157 369 158 370 159 371 160 372 161 373 162 374 163 375 376 164 165 377 166 167
378 379 168 169 380 381 170 382 171 172 383 173 384 385 174 175 386 176 387 177 178 388
179 389 180 390 181 391 182 392 183 393 394 184 395 185 396 186 397 187 398 188 399 189
400 190 191 192 193 194 195 196 197 198 199 200 All threads are running simultaneously
```