MD. EBRAHIM
ID: IT-22038

```
public class MiddleSquarePRNG {
private static volatile long seed = 675248;
private static synchronized long nextSeed (){
 long squared = seed * seed;
 String squaredStr = String.format("%012d", squared
 String middle = squaredStr. substring (3,9);
 seed = Long. parseLong (middle);
 return seed; }
}
public static int generate (int min, int max) {
 long num = nextSeed ();
return min+ (int) (num %(max-min+1)); }
public static float generate (float int, float max) {
 long num = nextSeed ();
 return min+ (num % 1000000)/1000000.0f *
                                         (max-min);
}
```

```
public static double generate (double min, double max){
  long num = nextSeed ();
  return min+(num %100000) /1000000.0 * (max-min);
  }
 static class RandomThread extends Thread {
  string type;
  public RandomThread (string type) {
  This. type = type; }
 public void run () {
 System.out.println ("Thread [" +type+ "] started:");
 for (int i=0; i<5; i++){
 switch (type) {
case "int" → System.out.println ("[int]"+generate (1,100));
case "float" → System.out.prinln ("[Float]"+generate (1.0f, 10.0f));
Case "Double" → System.out.println ("[double]"+generate
                                      (1.0, 10.0)); }
Try {
      Thread. sleep(100); }
catch (InterruptedException e) { }
system.out. println (" Thread [" + type + "] finished.");
public static void main [string[] args) {
Thread t1 = new RandomThread (" int");
Thread t2 = new RandomThread (" float");
Thread t3 = new RandomThread (" double");
 t1. start ();
 t2. start ();
 t3. start ();
        }
         }
```