

# Intro

A java file that we need to get the flag from.

## Flag

```
picoCTF{700l1ng_r3qu1r3d_2bfe1a0d}
```

## Solution

First we can start by getting the file type of the downloaded file by running `file KeygenMe.class` and we got

```
KeygenMe.class: compiled Java class data, version 55.0  
(Java SE 11)
```

## Running the file

Since this a java file, we can just run it by `java KeygenMe` and it prompts a key.

```
$ java KeygenMe  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
Enter key:  
kjfi  
Invalid key
```

If the key is invalid it just spits this out and exits.

## Analyzing the binary

After loading, analyzing and decompiling this file into `ghidra`, I found the main function checks on the flag character by character!

```
Decompile: main_java.lang.String[]_void - (KeygenMe.class)

188 }
189 cVar3 = objectRef.charAt(6);
190 if (cVar3 != 'F') {
191     pPVar1 = System.out;
192     pPVar1.println("Invalid key");
193     return;
194 }
195 cVar3 = objectRef.charAt(5);
196 if (cVar3 != 'T') {
197     pPVar1 = System.out;
198     pPVar1.println("Invalid key");
199     return;
200 }
201 cVar3 = objectRef.charAt(4);
202 if (cVar3 != 'C') {
203     pPVar1 = System.out;
204     pPVar1.println("Invalid key");
205     return;
206 }
207 cVar3 = objectRef.charAt(3);
208 if (cVar3 != 'o') {
209     pPVar1 = System.out;
210     pPVar1.println("Invalid key");
211     return;
212 }
213 cVar3 = objectRef.charAt(2);
214 if (cVar3 != 'c') {
215     pPVar1 = System.out;
216     pPVar1.println("Invalid key");
217     return;
218 }
219 cVar3 = objectRef.charAt(1);
220 if (cVar3 != 'i') {
221     pPVar1 = System.out;
222     pPVar1.println("Invalid key");
223     return;
224 }
225 cVar3 = objectRef.charAt(0);
226 if (cVar3 != 'p') {
227     pPVar1 = System.out;
228     pPVar1.println("Invalid key");
229     return;
230 }
```

Here, you can see that it checks if the 0th char is a **p** and the 1st char to be **i** and the third to be **c**... etc. This implies that it will be **picoCTF{...}**

## Extracting the key

After concatenating these letters, we will get **picoCTF{700l1ng\_r3qu1r3d\_2bfe1a0d}** and it is valid!