

Week 2 Challenge

TOTAL POINTS 5

1. A class type called "Pair" has already been defined for you. You need to write a function called pairFactory that creates an instance of Pair on the heap. Do not create the object on the stack. Then, your function needs to return a pointer to that created object.

5 points

Tips:

- An "instance of" a class type means one variable that is created with this type. We talk this way to clearly distinguish between the class type itself and specific variables that have this class type. For example, if you declare "Pair p;" then p is an instance of Pair. Also, it is common to say that an instance of a class is an "object".

- If you are stuck, please review: How do you create a variable on the stack? How do you create one on the heap?

- If you see the server message "InfraError," it means your program crashed with a segfault, typically. In this case, it's likely you returned an invalid address. Make sure you are creating the Pair on the heap.

- Pointers are variables that store memory addresses as their actual value. The pointer is said to "point to" whatever is located at the address that it stores. Returning a pointer means returning a copy of the pointer itself, that is, the address that the pointer stores, not a copy of the thing that is pointed to.

- Pointer variables are also located at addresses in memory, of course, but the address where the pointer is located is different from the address that it contains as its value.

```

1  #include <iostream>
2
3  // This class Pair has already been defined for you.
4  // (You may not change this definition.)
5  class Pair {
6  public:
7      int first, second;
8      void check() {
9          first = 5;
10         std::cout << "Congratulations! The check() method of the Pair class \n has executed. (Bu
11     }
12 };
13
14 // Insert your declaration and implementation of the function pairFactory()
15 // below, by replacing "..." with proper C++ code. Be sure to declare the
16 // function type to return a pointer to a Pair.
17
18 Pair * pairFactory() {
19     // ...
20     // (You can use as many lines as you want.)
21     Pair *ptr = new Pair;
22     return ptr;
23 }
24
25 // Your function should be able to satisfy the tests below. You should try

```

```
26 // some other things to convince yourself. If you have a bug in this problem,
27 // the usual symptom is that after you submit, the grader will crash with a
28 // system error. :-)
29 int main() {
30     Pair *p;
31     p = pairFactory();
32
33     // This function call should work without crashing:
34     p->check();
35
36     // Deallocating the heap memory. (Assuming it was made on the heap!)
37     delete p;
38
39     std::cout << "If you can see this text, the system hasn't crashed yet!" << std::endl;
40
```

[Run](#)[Reset](#)

Congratulations! The check() method of the Pair class has executed. (But, this isn't enough to guarantee that your code is correct.)

If you can see this text, the system hasn't crashed yet!