CSCI 112

Programming with C

Chapter 7, section 7.7

Enumerated Types

Test 2 Solutions

int arr[10] = {1};
ansi standard says that the rest of the elements in the array are set to 0;
But
int arr[10]; // if not global or static
you don't know what is stored.
Gave everyone credit if they selected the choice for 1, 0 for that question and then I removed that choice Wed afternoon.
BUT BAD PRACTICE TO ASSUME – do a test to check

Program 2 description

User Defined Types

- Up until now, we have only looked at four basic types in C (along with pointers and arrays)
- We can actually define our own data types using the typedef keyword
- An example of a type definition is:

```
typedef int counter_t ;
```

- This code defines a new type that is the same as an int
- The following declarations are now both equivalent (and legal):

```
int flag = 0;
counter_t flag = 0;
```

Enumerated Types

- It is common practice to assign numerically increasing integer values to a set of variables belonging to a typdef, so C gives us the option to use enum

Chapter 7, pgs 409-414

```
enum DAYS{
        sunday ,
        monday ,
        tuesday ,
        wednesday ,
        thursday ,
        friday ,
        saturday
};
```

- Now sunday through saturday have the values 0 through 6

Combining Enum and Typedef

```
typedef enum Day_t {
        sunday ,
        monday ,
        tuesday ,
        wednesday ,
        thursday ,
        friday ,
        saturday
};
```

Benefits of Enumerated Types

- Now we can do things like this:

```
Day_t today , tomorrow ;
   // assign value to today
   if ( today == saturday ) {
      tomorrow = sunday ;
   }
   else {
       tomorrow = today + 1;
}
```

Look at examples:
typedef.c
enum.c
enum1.c in
/public/examples/chap7

- And this:

```
day_t today ;
for ( today = monday ; today <= friday ; today ++) {
   // do something
}
```

Practice

- Pg 415, section 7.7
- #1, #2

- Solution for #1

1a. 0; 1b. 3; 1c. 0; 1d. Friday; 1e. Wednesday; 1f. 1