CS 161
Intro to CS I

1-D Arrays, Strings, and Command Line Arguments
Chap. 5
What is n1 and n2 after doStuff?

```c
void doStuff(int parValue, int &parRef)
{
    parValue = 100;
    cout << "parValue in call to doStuff = " << parValue << endl;
    parRef = 222;
    cout << "parRef in call to doStuff = " << parRef << endl;
}

int main()
{
    int n1 = 1, n2 = 2;
    doStuff(n1, n2);
    return 0;
}
```
Why does this fail?

void swap(int &lhs, int &rhs)
{
    lhs = rhs;
    rhs = lhs;
}

C++ Pass by Value

void swap(int, int);
int main() {
    int a = 5, b = 10;
    swap(a, b);
    cout << "a: " << a << " b: " << b;
}
void swap(int x, int y) {
    int temp = x;
    x = y;
    y = temp;
}
• What if we didn’t have temp?
C++ Pass by Reference

void swap(int &, int &);
int main() {
    int a=5, b=10;
    swap(a, b);
    cout << "a: " << a << " b: " << b;
}
void swap(int &x, int &y) {
    int temp = x;
    x = y;
    y = temp;
}
C++ Pointers

```cpp
void swap(int *, int *);  
int main() {
  int a=5, b=10;  
  swap(&a, &b);  
  cout << "a:" << a << "b:" << b;
}
void swap(int *x, int *y) {
  int temp = *x;  
  *x = *y;  
  *y = temp;
}
```
Variables vs. References

• Value Semantics
  – Values stored directly
  – Copy of value is passed
    ```
    int i, j=2;
    i=j;
    ```

• Reference Semantics
  – References to the value are stored
  – Copy of address is passed
    ```
    int *i, j=2;
    i=&j;
    ```
More about pointers...

```cpp
#include <iostream>

using std::cout;
using std::endl;

int main() {
    int *x;    // x variable holds pointer to int value
    int y=10;  // y variable holds int value, and gets 10
    x=&y;      // assign contents of x the address of y
    cout << "\n Y Addr: " << &y << " Value: " << y << endl;
    cout << "\n X Addr: " << &x << " Value: " << x << endl;
    cout << "\n X Value: " << *x << " Address of *X" << &x << endl;
    *x=20;     // assign pointer value
    
    // what if we assign address to pointer?
    //*x=y;
    
    // what if we assign the contents to the variable?
    // x=y;
    
    return 0;
}
```
What if we don’t have the y?

- We need to create the address space.
- How do we do this?
  - **new** type;
- For example:
  ```
  int *x;
  x = new int; //new returns an address
  *x = 10;
  ```
- [http://cslibrary.stanford.edu/104/](http://cslibrary.stanford.edu/104/)
What is an Array?

• **Array (ar·ray) n.** An ordered arrangement of related items.
  – Example: Array of colors in a rainbow.
    • Related items?
    • Ordered arrangement?
  – Class examples?
  – Computer Science
    • Same data type/data structure
    • Contiguous memory locations
Create 1-D Array

```c
int student_grades[5];
```

- How do you access each item?
- What does the array name represent?
- Why is the array name the address of 1\textsuperscript{st} element?
- What are the initial values?
Initialize/Assign Values

• **Declaration**
  
  ```c
  int student_grades[5] = {0, 0, 0, 0, 0};
  ```

• **Individual Elements**
  
  ```c
  student_grades[0]=0;
  ...
  student_grades[4]=0;
  ```

• **Why is this incorrect?**
  
  ```c
  student_grades={0, 0, 0, 0, 0};
  ```
Initialize/Assign Values...

• Using a Loop
  While Loop Example:
  i=0;
  while (i<5) {
    student_grades[i]=0;
    i++;
  }

  For Loop Example:
  for(i=0; i<5; i++)
    student_grades[i]=0;

• Which is better to use with arrays and why?
Read/Print 1-D Array Values

• Read Values From User
  
  for(i=0; i<5; i++) {
    cout << "Enter final grade for student: ";
    cin >> student_grades[i];
  }

• Print Values
  
  for (i=0; i<5; i++) {
    cout << "Student\'s final grade is " << student_grades[i] << endl;
  }

OSU Oregon State University
```cpp
#include <iostream>
#include <cstring>

using std::cout;
using std::cin;
using std::endl;

void print_intro(char name[]){
    cout << "Hello, " << name << endl;
}

int main(){
    char name[20];

    cout << "Enter your name: ";
    cin.getline(name, 20);

    cout << "Address of name & 1st element: " << (void *)name
    << " " << (void *)&name[0] << endl;
    cout << "Address of 2nd and 3rd elements: " << (void *)&name[1]
    << " " << (void *)&name[2] << endl;
    cout << "First letter in name and addr: " << name[0] << endl;
    cout << "Length: " << strlen(name) << endl;
    //How do we print the last letter?
    cout << "Last letter in name and addr: " << name[strlen(name)-1] << endl;
    //What happens when we step off the array bounds?
    //cout << "What if we go a little off: " << name[21] << endl;
    //cout << "What if we go way too far: " << name[100000] << endl;

    print_intro(name);
}``
Quiz #5

• How are you going to remove the non-alphabet characters from the sentence to check for a palindrome?
• Example: Madam, I’m adam