# Iteration 1: Implement Fundraising Project

## Celebrate and reflect on your project learnings.

Explain as if you are teaching someone else

- To give a sufficiently detailed report about how something works, about how elements in a system interact;
- To expound in  words, illustrations and examples that provide detail of your understanding.

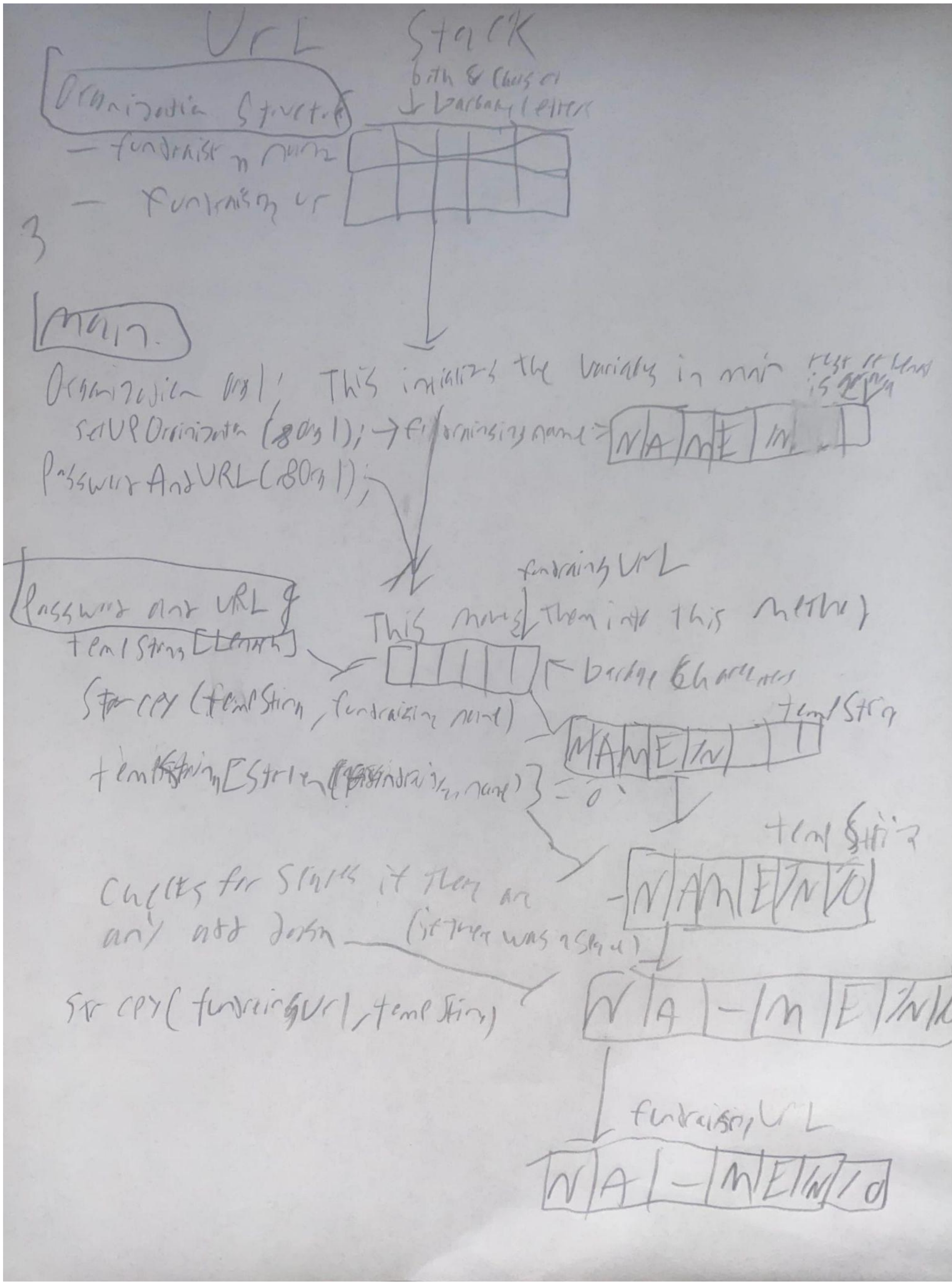
1. **Explain** your implementation to create the URL. Include explanations about how strings are stored, how they are passed and updated in a function. Include code examples in your explanation and drawings showing function stacks and how they are stored in memory.

   What i did to create my url is to first add the variable fundrasingURL into my fundraising structure, I then set its length to the constant LENGTH which i set earlier to 80, it should look like this: char funrasingURL[LENGTH]; Than in my function called getPasswordAndUrl, the structure is passed by reference using the *derfencer in its prototype and its actual method, than when the method is called, it has the structure name(org1) with the & in front of it. The prototype should look like  void passwordAndURL(Organization* org); and the call passwordAndURL(&org1);. Than in the function I created a temporary char array called tempString which is set to the same length as the url variable. Than I use the function strcpy to copy the fundraiser name into the tempString. I than add a end character to remove all garbage letters beyond the actual name.  Than I use a for loop to iterate throughout each character in the tempString and if any character is = to ‘ ’ than its replaced with ‘-’. Than i use strcpy again to add the string to the fundrasingURL variable. Here is an example:

```
 //Adds the name into a temp array
  strcpy(tempString, org->fundrasierURL);

 tempString[strlen(org->organizationName)] = '\0';
int length = strlen(tempString);
 //Replaces all spaces with -
    for (int i = 0; i < length; i++) {
        if (tempString[i] == ' ') {
            tempString[i] = '-';
        }
 }
// Add null terminator to the end of the tempString array
tempString[length] = '\0';
 // Copy tempString to fundraiserURL
 strcpy(org->fundrasierURL, tempString);
```

Here is a drawing for how this all interacts on the stack:


Url
Stack
Organization Structure
— fundraising name
— fundraising url
main
Organization org1; This initializes the variable in main
setUpOrganization(&org1); → fundraising name =
PasswordAndURL(&org1);
PasswordAndURL {
tempString[Length]
This moves them into this method
fundraisingURL
backing characters
strcpy(tempString, fundraising name)
tempString[strlen(fundraising name)] = '\0'
tempString
Checks for spaces if there are any add dash
(if there was a space)
strcpy(fundraisingUrl, tempString)
fundraisingURL

2. **Explain** a structure. Include how and when to pass a structure by reference or pass by value. Include code examples in your explanation for pass by reference.

a. A structure is a collection of variables that can be accessed throughout your program once initialized. It is usually initialized in main than passed by reference into methods through main when you want to edit its values, or pass by value when you want to display them. When creating the actual structure it is done using the type def function, and the variables inside the structure are only created and not set to any particular value. Here is an example of my structure in my implementation project.

```
typedef struct {
    char organizationName[LENGTH];
    char fullName[LENGTH];
    char fundraiserPurpose[LENGTH];
    char repName[LENGTH];
    char repPassword[LENGTH];
    char repEmail[LENGTH];
    double goalAmount;
    char fundrasierURL[LENGTH];
    double donationTotal;
    unsigned int amountOfDonors;
    double totalProcessingFees;
} Organization;
```

Here is an example of passing the structure by reference

```
void setUpOrganization(Organization* org1);
setUpOrganization(&org1);
```

And by value

```
void displayOrganizationCreation(Organization org);
displayOrganizationCreation(org1);
```

3. **Explain** what it means to implement maintainable and extendable code. Include at least 2 different examples from your code.
   a. What it means to implement maintainable and extendable code is to make you r code modularized so certain aspects can be worked on and updated at separate times, and avoid hard coding so other potential customers can change certain things to fit them easily.
      i. Modularized example:
         1. void setUpOrganization(Organization* org1);
         2. void displayOrganizationCreation(Organization org);
         3. void emailValid(Organization* org);
         4. void passwordAndURL(Organization* org);
         5. void organizationIntro(Organization org);
         6. bool getDonation(Organization* org, Donor* don);
         7. void finalSummary(Organization org);
         8. void getName(Donor* don);
         9. void getZip(Donor* don, Organization* org);
         10. void displayReceipt(Donor don, Organization org);
      ii. Easily changed and updated example with password
         1. #define CAPSMIN 1

2. #define LOWERMIN 1
3. #define NUMBERMIN 1
4. #define PASSWORDLENGTH 7
5. This way if there is any aspect of password creation another client wants (like longer password miniums or multiple numbers) it can be easily changed.

4. Share two strengths of yours for iteration 01 solution and at least one area you would like to improve on for iteration 02. Include information on how your project plan worked or didn't work.
    a. Strengths
        i. Almost everything is its own separate function, making the program very easily extendable
        ii. The only thing hardcoded is the zipcode length (which is the same everywhere) which allows this product to be very easily adapted to other clients.
    b. Ways to improve
        i. Work on accidental double triggers of puts (sometimes the code will run through an iteration without prompting the user when it should, happens ver rarely and in places where the amount of iterations is irrelevant, but this causes things such as "Enter donation" "Incorrect input please try again" "Enter donation" all without prompting the user until the second statement)
        ii. Try to be more efficient with code, my program, along with every program can have less redundant code which I hope to improve on in iteration 2.