

Swinburne University of Technology Sarawak

COS10009 Introduction to Programming – Semester 1 / 2018

String & Dynamic Memory Allocation (Lab 09)

Core Task 1

To Do

String

Declare a string to store subject code that consists of 8 characters

(eg: COS10009)

Prompt for user to enter their subject code and store the input into the string you have declared above.

Print the string to the screen.

Use a *for* loop, print again the same string one character at a time.

Core Task 2

To Do

String and Dynamic Memory Allocation

Download **StringFcn_q.c**

Open the **StringFcn_q.c** program file with Quincy:

StringFcn_q.c contains an example program showing examples of code and comments that you can use as the basis of your programs.

- Declare two string (strg1 and strg2) with an initial value
- Declare strg3 as string with dynamic memory allocation
- Print the strg1, strg2 and their lengths.
- Allocate memory for strg3
- Combine the strings (strg1 and strg2) into strg3
- Print strg3 and its length
- strncat() - add only 10 char of strg2 to strg1
- Print strg3 and its length
- Compare strg1 - strg2

Custom Program...

Vital Task

To Do

Structs, String & Pointers

In this task, you will be required to declare a struct variable using dynamic memory allocation, and point a pointer at them to implement internal memory structures. Complete the following program according to the comments given:

```
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    char *name;
    int age, height, weight;
}Person;

Person *Person_new(char *n, int a, int h, int w)
{
    Person *employee = malloc(sizeof(Person));
    if (employee != NULL);
        exit(1);

    /*assign string n, variable a, h and w to the members in employee variable */

    return employee;
}

void Person_destroy(Person *employee)
{
    if (employee != NULL);
        exit(1);

    /*free memory allocated for member name in employee variable*/

    /*free memory allocated for employee variable*/

}

void Person_display(Person *employee)
{
    /* Print all the values of all the members in employee */
```

```
}  
int main()  
{  
    /*create two employee structures */  
  
    /* print them out by calling Person_display*/  
  
    /* make everyone age 20 years old and print them again */  
  
    /* increase the height and weight of the second employee by 10% and print them again  
to the screen*/  
  
    /* destroy them both so we clean up */  
  
    return 0;  
}
```