

CS101 - Group Discussion 3

Instructions: As a group, work together to answer all the questions on the worksheet.

Discuss your approaches and make sure everyone understands the solution.

1. What is the output of the following program?

```
#include <stdio.h>
int f1(int val) {
    return val += 10;
}
int f2(int *ptr) {
    return *ptr += 10;
}
int main() {
    int a = 20, b=20;
    int r1 = f1(a), r2 = f1(a), r3 = f2(&b), r4 = f2(&b);
    printf("a=%d b=%d r1=%d r2=%d r3=%d r4=%d", a, b, r1, r2, r3, r4);
    return 0;
}
```

2. Circle syntactical and/or logical error(s) in the following code and correct them.

```
#include <stdio.h>
#include <string.h>
char* create_greeting(char *name) {
    char prefix[] = "Welcome, ";
    char result[50];
    strcpy(result, prefix); //copies content of 2nd arg to 1st arg
    strcat(result, name); //appends content of 2nd arg to 1st arg
    return result;
}
int main() {
    char *msg = create_greeting("student");
    printf("%s\n", msg);
    char *s1 = "CS101";
    char s2[5] = "^_^";
    strcpy(s2, s1); //copies content of 2nd arg to 1st arg
    printf("%s\n", s2);
    if (s1 = s2) printf("Strings are identical.\n");
    return 0;
}
```

3. What is the maximum number of active **puzzle** function calls on the call stack at any single moment? How many times is this peak achieved? Write a function, **int max_stack_depth(int n)**, that takes a non-negative integer **n** and returns the maximum number of active **puzzle** function calls that will be on the call stack when **puzzle(n)** is executed. Is there a mathematical closed form solution to this?

```
#include <stdio.h>
int puzzle(int n) {
    if (n <= 0) return 0;
    return puzzle(n - 1) + puzzle(n - 2);
}
int main() {
    puzzle(4);
    return 0;
}
```