# Department of Computing

# CS110: Fundamental of Computer Programming

# Class: BSCS-5C

# Lab 12: Arrays, Pointers, Functions

# Date: 04-01-2017

# Time: 10:00 to 01:00 & 2:00 to 5:00

# Instructor: Dr Anis Ur Rehman

# Lab 12: Arrays, Pointers, Functions

# Introduction

A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions. You can divide up your code into separate functions.

**Objectives**

The aim of this lab is to introduce basics of Functions which are most important aspect of C language. In the next lab some other topics related to Functions will be covered.

**Tools/Software Requirement**

Microsoft Visual Studio 2013

**Description**

**Example**

#include<stdio.h>

**void** area(); // Prototype Declaration

**void** main()

{

area();

}

**void** area()

{

**float** area\_circle;

**float** rad;

printf("\nEnter the radius : ");

scanf("%f",&rad);

area\_circle = 3.14 \* rad \* rad ;

printf("Area of Circle = %f",area\_circle);

}

**Out Put**

Enter the radius : 3

Area of Circle = 28.260000

Consider main function –

**void** main()

{

area();

}

We have just called a function , we can see that there is no variable or anything specified between the pair of round brackets.

**void** area();

Now in the prototype definition (line No 3) of the function we can see the return value as Void. Void means it does not return anything to the calling function.

**Lab Task**

1. Write a C program to check whether a number entered by user is prime or not using functions with no arguments and no return value.
2. Write a function minimum ( ) that shows the smallest of three floating-point numbers.
3. An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number, because 6 = 1 + 2 + 3. Write a function is Perfect() that determines whether parameter number is a perfect number. Display the factors of each perfect number to confirm that the number is indeed perfect.
4. Write a C program to find sum of first n natural numbers. Implement sum using functions. Print the result inside the function sum ( ). Note: Positive integers are known as natural number i.e. 1, 2, 3....n.
5. Remove the errors from the given program.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int pattern()

int main()

{

Int Pattern();

Patern();

getchar();

getchar();

return 0;

}

void Patern();

{

int i, space, rows, k = 0, count = 0, count1 = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i)

{

for (space = 1; space <= rows - i; ++space)

{

printf(" ");

++count;

}

while (k != 2 \* i - 1)

{

if (count <= rows - 1)

{

printf("%d ", (i + k));

++count;

}

else

{

++count1;

printf("%d ", (i + k - 2 \* count1));

}

++k;

}

count1 = count = k = 0;

printf("\n");

}

**}**

1. Transform the following code using functions

/\*

C Program to Display the ATM Transaction

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

unsigned long amount = 1000, deposit, withdraw;

int choice, pin, k;

char transaction = 'y';

void main()

{

while (pin != 1520)

{

printf("ENTER YOUR SECRET PIN NUMBER:");

scanf("%d", &pin);

if (pin != 1520)

printf("PLEASE ENTER VALID PASSWORD\n");

}

do

{

printf("\*\*\*\*\*\*\*\*Welcome to ATM Service\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("1. Check Balance\n");

printf("2. Withdraw Cash\n");

printf("3. Deposit Cash\n");

printf("4. Quit\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?\*\n\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("\n YOUR BALANCE IN Rs : %lu ", amount);

break;

case 2:

printf("\n ENTER THE AMOUNT TO WITHDRAW: ");

scanf("%lu", &withdraw);

if (withdraw % 100 != 0)

{

printf("\n PLEASE ENTER THE AMOUNT IN MULTIPLES OF 100");

}

else if (withdraw >(amount - 500))

{

printf("\n INSUFFICENT BALANCE");

}

else

{

amount = amount - withdraw;

printf("\n\n PLEASE COLLECT CASH");

printf("\n YOUR CURRENT BALANCE IS%lu", amount);

}

break;

case 3:

printf("\n ENTER THE AMOUNT TO DEPOSIT");

scanf("%lu", &deposit);

amount = amount + deposit;

printf("YOUR BALANCE IS %lu", amount);

break;

case 4:

printf("\n THANK U USING ATM");

break;

default:

printf("\n INVALID CHOICE");

}

printf("\n\n\n DO U WISH TO HAVE ANOTHER TRANSCATION?(y/n): \n");

fflush(stdin);

scanf("%c", &transaction);

if (transaction == 'n' || transaction == 'N')

k = 1;

} while (!k);

printf("\n\n THANKS FOR USING OUT ATM SERVICE");

}

1. Write a function called biggestEntry( ) that uses a two dimensional array and two parameters representing the row and column capacities. The function should return the value of the biggest entry in the array. For example, a program that uses the function biggestEntry follows.

int main()

{

int x[2][3] = {{1,2,3},{4,7,3}};

biggestEntry(x, 2, 3)

It should print 7 (since 7 is the biggest entry in the array)

**Deliverables**

Submit a document of MS word with code and snapshot of the output.