

PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES

***Computing Fundamentals & Programming***

**FALL 2020**

**Laboratory Exercise-15**

**Department: Physics**

**Name: Umar Shifaqat**

**Serial No. 43**

**Roll No. BS-20-GB-100864**

**Date: January 05, 2021**

**Home Task 1**

**Declare a structure, “Rect” you can define any number of members in the structure however the structure should must have members:**

**-> length (integer type)**

**-> width (integer type).**

**Try the following concepts:**

**(A) Declare two variables r1 and r2 of type Rect with the declaration of Rect,**

**Q: What is the scope of r1 and r2?**

**ASNWER:**

r1 and r2 are GLOBAL.

**Q: What is the longevity / life of r1 and r2?**

**ANSWER:**

Life of r1 and r2 is the life of the program. It remains till the end of program.

**Q: We have not created the variables in main but we have used them in main and program did not raise any error, why?**

**ANSWER:**

Because they were declared before main with the structure. They are not local to main.

**(B)Now declare the variables r1 and r2 in main (remove the previous declaration of r1 and r2) and execute the program again.**

**Q: What is the scope of r1 and r2 now?**

ANSWER:

Now, r1 and r2 are “Local to main”.

**Home Task 2**

**(A)**

Write a simple function displayRect(Rect rect) that displays the values of member of rect. Modify your above program and display the value of r1 and r2 using displayRect function.

**INPUT**

#include<stdio.h>

struct Rect

{

int length;

int width;

}r1,r2;

void displayRect(Rect rect);

void main()

{

r1.length=10;

r1.width=30;

r2.length=25;

r2.width=40;

printf("For r1:");

displayRect(r1);

printf("\nFor r2:");

displayRect(r2);

getchar();

getchar();

}

void displayRect(Rect rect)

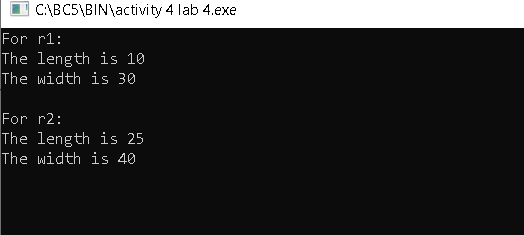
{

printf("\nThe length is %d",rect.length);

printf("\nThe width is %d\n",rect.width);

}

**OUTPUT**

****

**(B)**

Write a function “int isSquare(Rect rect) ”:

The function will return 1 if the rect members length and width have same values otherwise the function should return 0.

Modify your program and call this function to check that r1 or r2 are squares or not, print suitable messages.

**INPUT**

#include<stdio.h>

struct Rect

{

int length;

int width;

}r1,r2;

int isSquare(Rect rect);

void main()

{

int result;

r1.length=30;

r1.width=30;

r2.length=25;

r2.width=40;

result=isSquare(r1);

(result==1)?puts("\tR1 is square"):puts("\tR1 is not square");

result=isSquare(r2);

(result==1)?puts("\tR2 is square"):puts("\tR2 is not square");

getchar();

getchar();

}

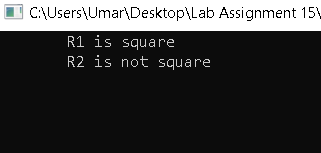
int isSquare(Rect rect)

{

return (rect.length==rect.width)?1:0;

}

**OUTPUT**

****

**(C)**

Write a function “int compareRect(Rect rect1, Rect rect2, )”: The function should compare the length of rect1 with length of rect2 and width of rect1 with width of rect2, if length and width of rect1 and rect2 are same the function should return 1 otherwise function should return 0.

Modify your program to compare r1 and r2, display proper messages.

**INTPUT**

#include<stdio.h>

struct Rect

{

int length;

int width;

}r1,r2;

int compareRect(Rect rec1,Rect rec2);

void main()

{

int result;

r1.length=30;

r1.width=40;

r2.length=25;

r2.width=40;

result=compareRect(r1,r2);

(result==1)?puts("\tR1 and R2 are equal"):puts("\tR1 and R2 are not equal");

getchar();

getchar();

}

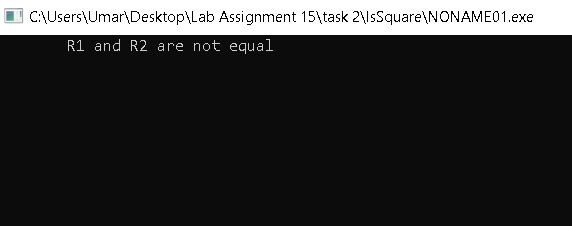
int compareRect(Rect rec1,Rect rec2)

{

return (rec1.length==rec2.length && rec1.width==rec2.width)?1:0;

}

**OUTPUT**

****

**THE END**