National University of Computer and Emerging Sciences



**Laboratory Manual**

***(Computer Programming)***

|  |  |
| --- | --- |
| Course Instructor | Sarim Baig |
| Lab Instructor(s) | Ahmad Raza  Waqas Manzoor |
| Section | A |
| Semester | Spring-2017 |
| Lab Date | 24-01-2017 |

Department of Computer Science

FAST-NU, Lahore

**Lab Manual (1)**

**(Debugging, Pointers and Structures)**

**Instructions:**

This is an individual Lab. You are NOT allowed to work/submit in form of group. Absolutely NO collaboration is allowed. Any traces of cheating would result in an “F” grade in this Lab.’

Keep the following good programming practices in mind when writing your code:

• Comment your code intelligently.

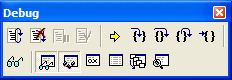
• Indent your code properly.

• Use meaningful variable names.

• Use meaningful prompt lines/labels for input/output.

**Problem 1:**

Open a new project in visual studio and type a simple program to find the maximum element in an array and compile it. Practice with the following debugging commands:

** **

|  |  |  |  |
| --- | --- | --- | --- |
| Short cut key | Icon | Menu | Explanation |
| F-9 |  |  | Insert/Remove breakpoint |
| F-5 |  | Debug-Go | Execute a program until the next breakpoint |
| Shift F-5 |  | Debug-Stop debugging | To stop debugging a program. It will stop executing the program |
| F-10 |  | Debug-StepOver | Go to the next statement |
| F-11 |  | Debug-Step Into | Go inside a function |
| Shift F-11 |  | Debug – Step Out | Come out of the function |
|  |  | Debug - Run to cursor | Execute all statements till the statement on which the cursor is placed or until the next breakpoint |
| Alt -3 |  | Debug-Windows-Watch | Show the window where only the variables in scope are shown |
| Alt-4 |  | Debug-Windows-Variables | Show the window in which you can type a variable name to see its value |
| Alt-7 |  | debug-windows-call stack | You can see the activation of stack of functions here |

**Problem 2:**

Introduce int variables x and y, initialize them with 5 and 10 respectively. Then create int\* pointer variables p and q. Then p to the address of x, and q to the address of y then swap the values of x and y using pointers (p and q).

Then print the following information:

(1) The address of x and the value of x.

(2) The value of p and the value of \*p.

(3) The address of y and the value of y.

(4) The value of q and the value of \*q.

(5) The address of p (not its contents!).

(6) The address of q (not its contents!)

**Problem 3:**

1. Declare two variables Sum1 and Sum2.
2. Declare two integer variables A and B and take input in them.
3. Declare three integer pointers Pointer1, Pointer2 and Pointer3.
4. Assign the address of variable A to pointer variable Pointer1.
5. Assign the address of variable B to pointer variable Pointer2.
6. Subtract the value of variable pointed to by Pointer1 from the value of variable pointed to by Pointer2 (You should not use variables A and B in this statement). Assign the result of Subtraction to variable Sum1.
7. Assign the value of pointer variable Pointer1 to pointer variable Pointer3.
8. Assign the value of pointer variable Pointer2 to pointer variable Pointer1.
9. Assign the value of pointer variable Pointer3 to pointer variable Pointer2.
10. Again subtract the value of variable pointed to by Pointer1 from the value of variable pointed to by Pointer2, and assign the result of subtraction to variable Sum2.
11. Print the sum of variables Sum1 and Sum2.

**Problem 4:**

1. Introduce 2 variables i (int), j (float) and initialize to 5, 1.5 respectively.
2. Introduce 2 variables xPtr (int\*), yPtr (float\*) and assign address of i, j respectively.
3. Print values of i, j and addresses in xPtr, yPtr.
4. xPtr ++; yPtr ++;
5. Print addresses in xPtr,yPtr.
6. xPtr --; yPtr --;
7. Print addresses in xPtr, yPtr.
8. xPtr = xPtr + 3; yPtr = yPtr + 4;
9. Print addresses in xPtr, yPtr.

**Problem 5:**

Declare three integers x, y and sum and three pointers xPtr, yPtr, sumPtr. Point three pointers to their respective variable.

Take input in x and y using xPtr and yPtr. **Do not use direct references i.e. x and y integers**

Add x and y and save the result in sum. **Do not use direct references i.e. x, y and sum integers**

Print addition’s result.

For example, if user entered x = 5 and y = 9 your program should print: 5 + 9 = 14.

**Do not use direct references i.e. x, y and sum integers**

Note: You have to do all the processing using pointers i.e. indirect references to variables.

**Problem 6: (Post Lab.. submit in next lab)**

**Write this code and compile it (Don’t Copy-Paste)**

## struct student

## {     int roll\_no;

## char name[20],city[20];

## int phone;

## };

## int main()

## {     student std;

## cout<<"Enter roll number?"<<endl;

## cin>>std.roll\_no;

## cout<<"Enter name of student?"<<endl;

## cin>>std.name;

## cout<<"Enter name of city of student?"<<endl;

## cin>>std.city;

## cout<<"Enter phone of student?"<<endl;

## cin>>std.phone;

## cout<<endl;

## cout<<"Roll no. : "<<std.roll\_no<<endl;

## cout<<"Name: "<<std.name<<endl;

## cout<<"City: "<<std.city<<endl;

## cout<<"Phone: "<<std.phone<<endl;

## }

**Allocate an array of 10 student variables. (Use the Structure declared above)**

***Remember:*** *Honesty always gives fruit (no matter how frightening is the consequence); and*

*Dishonesty is always harmful (no matter how helping it may seem in a certain situation)!*