**Department of Electrical and Computer Engineering, NSU**

**CSE 115L: Fundamentals of Computer Programming (Section 4)**

**Lab 13 (Pointers), Faculty: Rsl**

|  |  |
| --- | --- |
| **Example 1: accessing array elements using pointer** | **Example 2: passing array as function using pointer** |
| #include<stdio.h>  int main()  {  int a[4]={2,5,1,6};  int \*ptr=a;  //note that & is not required  int i;  for(i=0; i<4; i++)  {  printf("\*(ptr+%d) = %d same as a[%d]\n",i,\*(ptr+i),i);  }  /\* for(i=0; i<4; i++)  {  printf("\*(a+%d) = %d same as a[%d]\n",i,\*(a+i),i);  }\*/  return 0;  } | #include<stdio.h>  void printNum(int \*ptr, int len);  int main()  {  int a[4]={4,10,1,5};  printNum(a,4);  return 0;  }  void printNum(int \*ptr,int len)  {  int i;  for(i=0; i <len ; i++ )  {  printf("\*(ptr+%d) = %d \n",i,\*(ptr+i),i);  // printf("ptr[%d]= %d \n",i, ptr[i]);  }  } |

|  |  |
| --- | --- |
| **Example 3: returning pointer from functions** | **Key points to remember about pointers in C** |
| #include<stdio.h>  int\* getRandom();  int main () {  int \*p;  p = getRandom();  printf("address of %d\n",p);  printf("value at \*p : %d\n", \*p );  return 0;  }  int\* getRandom( ) {  static int j;  j=rand()%10;  printf("value of j: %d\n",j);  printf("Address of j: %d\n", &j);  return &j;  } | **1)** Normal variable stores the value whereas pointer variable stores the address of the variable.  **2)** The content of the C pointer always be a whole number i.e. address.  **3)** Always C pointer is initialized to null, i.e. int \*p = null.  The value of null pointer is 0.  **4)** & symbol is used to get the address of the variable.  **5)** \* symbol is used to get the value of the variable that the pointer is pointing to.  **6)** If pointer is assigned to NULL, it means it is pointing to nothing.  **7)** Two pointers can be subtracted to know how many elements are available between these two pointers. |

**Some Common Mistakes**

int c, \*pc;

pc=c; //pc is address whereas, c is not an address.

\*pc=&c; //&c is address whereas, \*pc is not an address.

Note:

**Equivalent Expression**

list[2]=5;

\*(p+2)=5;

p[2]=5;

\*(list+2)=5;

All of the above are same.

**Task (10 marks)**

**1.** Write a C program to calculate the area and perimeter of a circle using pointer.

**void areaPeri(int \*r)**



**2.** Generate the following series using the swap function in the examples and using pointers.

1 1 1 3 5 9…………………………..n, where n will generate that many terms.



**3.** Create an array of size given by the user. Find a number taken as input from user. Your task is to print the memory address and the index of the number using pointer to travel the array. Note that if number is found at multiple position you should print all memory address.

