**Practical 5**

1. Trace the programs below. Understand the output produced on the screen.
2. Use of an independent reference variable

// Use of an independent reference variable

#include <iostream>

using namespace std;

int main()

{

int i=13;

int &iref = i; //declaring a reference variable

cout << "The value is : " << iref << endl;

i--;

cout << "After decrementing : " << iref << endl;

iref = 99;

cout << "The value now is : " << i << endl;

return 0;

}

1. Pointer with *new* operator

#include <iostream>

using namespace std;

int main()

{

int \*p1, \*p2;

p1 = new int;

p2 = new int;

\*p1 = 10;

\*p2 = 20;

cout << \*p1 << " " << \*p2 << endl;

\*p1 = \*p2;

cout << \*p1 << " " << \*p2 << endl;

\*p1 = 30;

cout << \*p1 << " " << \*p2 << endl;

return 0;

}

#include <iostream>

using namespace std;

int main()

{

int number = 88;

int \* pNumber;

pNumber = &number;

cout << pNumber << endl;

cout << &number << endl;

cout << \*pNumber << endl;

cout << number << endl;

\*pNumber = 99;

cout << pNumber << endl;

cout << &number << endl;

cout << \*pNumber << endl;

cout << number << endl;

cout << &pNumber << endl;

return 0;

}

1. Pointer with character string

#include <iostream>

using namespace std;

int main()

{

char note[]="Silence is Golden";

char \*ptr;

ptr = note;

cout<<ptr<<endl;

cout<<++ptr<<endl;

note[11] = '\0';

cout<<note<<endl;

cout<<++ptr<<endl;

return 0;

}

1. Pointer arithmetic

#include<iostream>

using namespace std;

int main()

{

char a[] = "Hello", b[] = "Bye" , \*p1, \*p2, \*p3;

p1 = p2 = a;

p3 = b;

cout << "\*p1 = " << \*p1 << ", \*p2 = " << \*p2 << ", \*p3 = " << \*p3 << endl;

cout << "\*p1++ = " << \*p1++ << ", \*++p2 = " << \*++p2 << ", (\*p3)++ = "

<< (\*p3)++ << endl;

cout << "\*p1 = " << \*p1 << ", \*p2 = " << \*p2 << ", \*p3 = " << \*p3 << endl;

return 0;

}

1. Write a short C++ program that declares and initializes (to any value you like) a double, an int, and a char. Next declare and initialize a pointer to each of the three variables. Your program should then print the address of, and value stored in, and the memory size (in bytes) of each of the six variables. Use the ***sizeof*** operator to determine the memory size allocated for each variable.

* Note: you need to use *static\_cast <void \*>* to print the address of a char.

1. Given the following declarations, write a C++ program which prints the characters in a C string in a reverse order using pointer notation.

char s[10] = "abcde";

char\* cptr;