**CS-103**

**Autumn 2013**

**Tut 6 (Pointers using C++)**

**Q1. Enter the following C++ code. Run the program and explain the results. Also draw a diagram illustrating the relationships between the different variables in the given program.**

#include <iostream>

using namespace std;

int main ()

{

int n = 55;

int\* p= &n; // p holds the address of n

int \* q;

cout <<"n = "<< n<<endl;

cout<<”\*p= “<<\*p<<endl;

cout <<"&n (address of n) = "<<&n<<endl;

cout<<"p - value of pointer variable p (i.e address of n) = " <<p<< endl;

cout<<"&p - address of pointer varibale = "<<&p<<endl;

cout << "&(\*p) - address of dereferenced pointer p = "<< &(\*p) << endl<<endl;

q = &n;

cout<<"q - value of pointer variable q (i.e address of n) = " <<q<< endl;

cout<<”\*q= “<<\*q<<endl;

system("pause");

return 0;

}

**Q2. Enter the following code. Observe the output.**

#include<iostream>

using namespace std;

int \*ptr;

void testFunction()

{

int n=55;

ptr = &n;

}

int main()

{

testFunction();

cout<<"Value of \*ptr"<<\*ptr<<endl;

system("pause");

return 0;

}

1. Compile and run the program. If the result is incorrect, explain why the code fails although there is no compilation error.
2. Modify the function in the above program to get the correct result. If the result is correct, explain why the result is OK.

Q3. WAP to function to swap two numbers using pointers with given function prototype

**void swap (float \*,float \*);**

Q4. WAP using pointer to display the elements of an 1D array with 6 elements (hint using any of the four arithmetic operators ++,--, +, - for pointer arithmetic)

1. in the same order
2. in the reverse order
3. Odd indexed elements
4. Even indexed elements

Q5. WAP to find the sum of all elements of given a 1D array **int a[ ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }** and prototype of the function **int sum(int \*, int )** where array as pointer and its size is passed.

Q6. WAP given a 2D array i.e. **int Array[3][3]={{1,2,3},{4,5,6},{7,8,9}}**

Display the elements in 3x3 form

1. without using pointer
2. using pointer
3. address of each element
4. using & reference operator
5. using pointer

Q 7. WAP using pointers for the following:

1. Create a Dynamic array having name marksArray with size 4
2. Enter marks in the array
3. Display the values of the elements of the array
4. Create another array of size twice than marksArray having name marksArrayA
5. Transfer marksArray elements to marksarrayA in position 0 to 3
6. Delete the dynamic array marksArray
7. Now enter marks in vacant elelments of marksArrayA. Note only elements 4 to 7 are vacant
8. Dispaly elements of marksArrayA
9. Delete dynamic array marksArrayA

**Also explain why deletion of array is necessary. If omitted, what would be it’s implication.**

Q8. WAP that takes the size n of an array from the user. It then dynamically creates an array of size n of type double. It should prompt the user to input n values (of type double) which are then stored in the array. Also write a recursive function sum whose prototype is double sum( \*a,int size) and it recursively finds the sum of the entire array. The function should finally return the sum which should be displayed by the main in your program.

Q9. Implement a class CPolygon in C++ as given below:

#include<iostream>

using namespace std;

class CPolygon {

protected:

int width, height;

public:

CPolygon() { width =0; height = 0; }

CPolygon(int w, int h) { width = w; height = h;}

int area() { return (width\*height);}

void display() { cout << this->area() << endl ;}

};

int main()

{

CPolygon \* ptr\_polygon1 = new CPolygon(2,3);

CPolygon \* ptr\_polygon2 = new CPolygon(4,5);

CPolygon poly1(6,7);

CPolygon poly2(8,9);

ptr\_polygon1->display();

ptr\_polygon2->display();

delete ptr\_polygon1;

delete ptr\_polygon2;

ptr\_polygon1= &poly1;

ptr\_polygon2= &poly2;

ptr\_polygon1->display();

ptr\_polygon2->display();

delete ptr\_polygon1;

delete ptr\_polygon2;

system("pause");

return 0;

}

1. Write the output of the above program and explain the statements in the main function.
2. Write the equivalent class in Java and write your observation for difference and similarity in C++ and Java implementations.