OOPS LAB
ASSIGNMENT-1

NAME : Swarnendu Banerjee

ROLL : 002311001016

DEPARTMENT : IT – A1

C++ style Input/Output

2. Write a program that accepts two integers from keyboard, adds them and prints their values. Use cin and cout.

CODE:

```
#include<iostream>
using namespace std;

int main(){
int pdt=1;
for(int i=1;i<=6;i++){
pdt*=i;
cout<<i<<"! ="<<pdt<<endl;}
return 0;}
```

3. Create a factorial table using cout as follows:

```
1! = 1
2! = 2
3! = 6
…
6! = 720
```

CODE:

```
#include<iostream>
using namespace std;

int main(){
int pdt=1;
for(int i=1;i<=6;i++){
```

```
pdt*=i;
cout<<i<<"! ="<<pdt<<endl;}
return 0;}
```

Flexible Variable Declaration

4. Write a program to print 1 to 10 using a for loop. Declare the loop variable inside the for loop. Check the scope of this variable.

   CODE:

```
#include<iostream>
using namespace std;

int main(){
int i=100;
for(int i=1;i<=10;i++)cout<<i<<"\n";
cout<<" out of the loop the value of i is :"<<i;
}
```

5. Write a program to display Celsius to Fahrenheit conversion table using a for loop. Consider only 0˚ to 100˚ Celsius. Declare variables when they are used for the first time.

   CODE:

```
#include<iostream>
using namespace std;

double conversion(int  c){
 return 9*c/5.0+32.0;}
```

```
int main(){
  for(int i=0;i<=100;i++){
  cout<<i<< " celcius is :"<<conversion(i)<<" Fahrenheit"<<endl;}
return 0;}
```

Constants

6. Write a program that defines a constant PI and takes radius of a circle from keyboard and prints area of that circle.

   CODE:

```
#include<iostream>
using namespace std;
#define PI 3.14

int main(){
double r;
cout<<"enter the radius of the circle : ";
cin>>r;
cout<<"the area of the circle is : "<<PI*r*r;
return 0;}
```

7. Write a function that takes an integer and returns the factorial of that number. Declare function parameter as const. Call the function with some argument from main function, store the result and print it.

CODE:

```
#include<iostream>
using namespace std;
```

```
int fact(const int n){
if(n<=1)return 1;
else return n*fact(n-1);}

int main(){
 int n;
 cout<<"enter the number :";
 cin>>n;
 int j=fact(n);
 cout<<"the factorial of the number is "<<j;
return 0;}
```

Reference

8. Write a function swap() that takes two integer arguments and interchanges the values of those arguments using reference. Now in the main function, instantiate two integer variables with some values. Print their values. Call the swap function with these variables. Finally print the values of those variables. Check the result.

CODE:

```
#include<iostream>
using namespace std;

void swap(int &a,int &b){
 int t=a;
 a=b;
b= t;}
  int main(){
  int a=2,b=3;
  cout<<"before swapping :"<<"a ="<<a<<" and b="<<b;
```

```
  swap(a,b);
  cout<<"after swapping:"<<"a ="<<a<<" and b="<<b;
return 0;}
```

9. Now write another function swap() that takes two strings (character array) and interchanges them without reference parameters. Test this function using some arguments. Rewrite the function using reference parameters. Again test this function with some arguments.

CODE:

```
#include<iostream>
using namespace std;
#include<cstring>
void swap(char*c,char*d){
char temp[100];
strcpy(temp,c);
strcpy(c,d);
strcpy(d,temp);}

int main(){
char arr1[100],arr2[100];
cout<< "enter the two strings :";
cin>>arr1>>arr2;
swap(arr1,arr2);
cout<<"after swaping the strings are :"<<arr1<<" and "<<arr2;
return 0;}
```

Constants and references

10. Write a function that takes an integer and returns the factorial of that number. Declare function parameter as read only reference. Call the function with some argument from main function, store the result and print it.

CODE:

```
#include<iostream>
using namespace std;

int fact(const int &n){
if(n<=1)return 1;
else return n*fact(n-1);}

int main(){
int j=4;
int fc=fact(4);
cout<<"factorial of "<<j<<"is "<<fc;
return 0;}
```

Constants and pointers

11. Write a function Strcpy to copy one string to another with suitable formal parameters declarations. Following points must be considered.
    a) Source string must not get modified
    b) Target string is allowed to get modified
    c) The Pointers must be constant pointers.

    Use it to copy some strings.

```
#include<iostream>
using namespace std;
```

```
void strcpy(const string &str,char*ch){
 int i;
for(i=0;i<str.length();i++)*(ch+i)=str[i];
*(ch+i)='\0';}

int main(){
 string s="JADAVPURUNIVERSITY";
 char ch[s.length()+1];
strcpy(s,ch);
cout<<ch;
return 0;}
```

Inline function

12. Write an inline function add() that takes three integer arguments and returns the sum of these arguments.

CODE:

```
 #include<iostream>
using namespace std;
inline int add(int a,int b,int c){
 return a+b+c;}

int main(){
 int a,b,c;
 cout<<" enter the three integers :";
cin>>a>>b>>c;
cout<<"sum of the integers are :"<<add(a,b,c);
return 0;}
```

Function Overloading

13. Consider the following two scenarios:
    a) We want to find out the maximum between three integers.

b) We also want to find out the maximum element of an array of integers.

Write two overloaded functions for these two scenarios.

CODE:

```
#include<iostream>
using namespace std;
int maxm(int a,int b,int c){
  if(a>b && a>c)return a;
  else if(b>c) return b;
  else return c;}

int maxm (int arr[],int n){
  int maxmi=arr[0];
  for(int i=1;i<n;i++){
  if(arr[i]>maxmi)maxmi=arr[i];}
  return maxmi;}

int main(){
  int a=2,b=3,c=6;
  int arr[]={1,2,3,4,5};
  cout<<"maxm when three numbers :"<<maxm(a,b,c)<<" maxm
when the integer array is passed :"<<maxm(arr,5);
return 0;}
```

14. Write two overloaded functions print() such that one prints the elements of a vector and the other prints elements of a matrix. Note that a vector

and a matrix may be represented as a one-dimensional array and a two-dimensional array respectively.

CODE:

```
 #include<iostream>
using namespace std;
#include<vector>

void print(vector<int>&v){
 for(int i=0;i<v.size();i++){
 cout<<v[i]<<" ";}}

void print(int arr[2][2]){
 for(int i=0;i<2;i++){
  for(int j=0;j<2;j++){
  cout<<arr[i][j]<<" ";}}}

int main(){
 vector<int>v(4,2);
 int arr[2][2]={9,8,7,6};
 print(v);
 cout<<endl;
 print(arr);
return 0;}
```

Default values for function parameters

15. Consider function add() in 13. Specify the default values for second and third parameters to 0 (zero). Now call this function with three, two and one arguments and see the result.

CODE:

```
#include<iostream>
using namespace std;

inline void add(int x,int y=0,int z=0){
  cout<< x+y+z<<endl;}

int main(){
  add(3,2,4);
  add(2,1);
  add(2);
 return 0;}
```