OOPS ASSIGNMENT – 1

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

```
#include <iostream>
using namespace std;
int main(){
   int m,n;
   cout<<"enter the two numbers: ";
   cin>>m>>n;
   int sum;
   sum=m+n;
   cout<<"m= "<<m<<"n= "<<n<<endl;
   cout<<"The sum is: "<<sum;
   return 0;
}
```

3. **Create a factorial table using cout as follows:**

1! = 1
2! = 2
3! = 6
…
6! = 720

```
#include <iostream>
using namespace std;
int main(){
   int n,f=1;
   cout<<"enter limit: ";
   cin>>n;
   for (int i=1;i<=n;i++){
      f=f*i;
      cout<<i<<"! = "<<f<<endl;
   }
   return 0;
}
```

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.**

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

```
int main()
{
  int i = 100; //initial value of i
  for(int i = 1; i <= 10; i++) // i declared within loop
  {
     cout << "Inside the loop i = " << i << "\n";
  }
   cout << "Initial value of i = " << i; // initial value of i is printed

  return 0;
}
```

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.**

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

int main()
{
  int c;
  double f;
  for(c = 0; c <= 100; c++)
  {
     f = (9.0/5.0) * c + 32;
     cout << "Celsius = " << c << " Fahrenheit = " << f << "\n";
  }
  return 0;
}
```

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

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

int main()
{
  const double PI = 3.1415926535;
  double r,a;
  cout << "Enter the radius : ";
```

```
  cin >> r;
  a = PI*r*r;
  cout << "\nArea of the circle = " << a;
  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.**

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

int fact(const int z)
{
   int f = 1;
   for(int i = 1;i <= z; i++)
   f*=i;
   return f;
}
int main()
{
  int n;
  cout << "Enter a number : ";
  cin >> n;
  int x = fact(n);
  cout << "The factorial of " << n << " = " << x;
  return 0;
}
```

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.**

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

void swap(int &x, int &y)
{
   x+=y-(y=x);
```

```
}
int main()
{
  int a,b;
  a = 10; b = 20;
  cout << "Initial value of a = " << a << "\n";
  cout << "Initial value of b = " << b << "\n";
  swap(a,b);
  cout << "Final value of a = " << a << "\n";
  cout << "Final value of b = " << b << "\n";
  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.**

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

void swapByAddress(string *x, string *y)
{
   string z = *x;
   *x = *y;
   *y = z;
}
void swapByReference(string &x, string &y)
{
   string z = x;
   x = y;
   y = z;
}
int main()
{
  string a,b;
  a = "Information"; b="Technology";
  cout << "Initial value of a = " << a << "\n";
  cout << "Initial value of b = " << b << "\n";
  swapByAddress(&a,&b);
  cout << "Final value of a swapping by address = " << a << "\n";
  cout << "Final value of b swapping by address = " << b << "\n";
  return 0;
}
```

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.**

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

int fact(const int &z)
{
   int f = 1;
   for(int i = 1; i <= z; i++)
   f*=i;
   return f;
}
int main()
{
  int n;
  cout << "Enter a number : ";
  cin >> n;
  int x = fact(n);
  cout << "The factorial of " << n << " = " << x;
  return 0;
}
```

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 = "InformationTechnology";
  char ch[s.length()+1];
  strcpy(s,ch);
  cout << ch;
  return 0;
}
```

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

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

inline int add(int a, int b, int c){
   return a+b+c;
}

int main()
{
  int x,y,z;
  cout << "Enter three numbers : ";
  cin >> x >> y >> z;
  cout << "\nSum of the three numbers = " << add(x,y,z);
  return 0;
}
```

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.

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

int Maximum(int a, int b, int c)
{
   return (a>b&&a>c)? a : (b>c) ? b : c;
}

int Maximum(int arr[], int n)
{
   int m = arr[0];
   for(int i = 1; i < n; i++)
```

```
    m = max(m,arr[i]);

    return m;
}

int main()
{
    int x,y,z;
    cout << "Enter 3 numbers : \n";
    cin >> x >> y >> z;
    cout << "Maximum of the three numbers is = " << Maximum(x,y,z) << "\n";
    int n;
    cout << "Enter the size of the array : ";
    cin >> n;
    int a[n];
    cout << "Enter the elements of the array : \n";
    for(int i = 0; i < n ;i++)
    cin >> a[i];
    cout << "Maximum number in the array is = " << Maximum(a,n);
   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.

```
#include <iostream>

using namespace std;

void print(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << "\n";
}

void print(int m, int n, int **matrix)
{
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
```

```
            cout << matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

int main()
{
    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int size = sizeof(arr) / sizeof(arr[0]);
    print(arr, size);

    int m = 3, n = 3, k = 1;
    int **matrix;
    matrix = new int *[m];
    for (int i = 0; i < m; i++)
    {
        matrix[i] = new int[n];
        for (int j = 0; j < n; j++)
        {
            matrix[i][j] = k++;
        }
    }
    print(m, n, matrix);
}
```

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.

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

inline void add(int a, int b=0, int c=0){
   cout << "Sum of the three values = " << a+b+c << "\n";
}

int main()
{

  add(2,3,1);
  add(1,2);
  add(1);
  return 0;
}
```