C++ Overloading ‘ABC’ 18-2-2023

1.	**All types Arrays read print - function overloading**
	Overload a function readArr(..) which reads either array of integers, or characters, or floats, or strings.
	Overload a function printArr(..) which prints any type of array.
	void swap(int &ix,int &iy);
	void swap(float &fx,float &fy);
	void swap(char &cx,char &cy);

2.	**All Swaps- function overloading**
	Overload swap function as shown below.
	void swap(int &ix, int &iy);
	void swap(float &fx, float &fy);
	void swap(char &cx, char &cy);
	void swap(int *x, int *y);
	void swap(float *fx, float *fy);
	void swap(char *cx, char *cy);

3.	**Matrices – operator overloading** Let M1 and M2 are two matrices. Overload operators + , – and ->* to find out M3=M1+M2 , M3 = M1-M2 , M2=M1->*x (x is an integer) The operator ->* takes a Matrix and an integer x, and multiplies each element of matrix with x.
4.	**Cstring – operator overloading** Create a class CString to represent a string. a) Overload the + operator to concatenate two strings. b) Overload the = = operator to compare 2 strings. c) Overload the ! operator to to reverse the case of each alphabet in the string
5.	**Structure input >> , output << overloading** Read Notes given below. Overload the operator >> for reading a structure variable S. The input statement should be cin>>S; Overload the operator << for printing contents of a structure variable S. The ouput statement should be cout<<S;
6.	**Class Objects input >> , output << overloading** Read Notes given below. Overload the operator >> for reading an object of your own class. The input statement should be cin>>obj1; Overload the operator << for printing contents of an object of your own class. The ouput statement should be cout<<obj1;

Notes :

Overloading the << Operator for Your Own Classes

Output streams use the insertion (<<) operator for standard types. You can also overload the << operator for your own classes.

Example

The write function example showed the use of a Date structure. A date is an ideal candidate for a C++ class in which the data members (month, day, and year) are hidden from view. An output stream is the logical

cout <<dt;

To get cout to accept a Date object after the insertion operator, overload the insertion operator to recognize an ostream object on the left and a Date on the right. The overloaded << operator function must then be declared as **a friend of class Date so it can access the private data** within a Date objec

```
// overloading the operator << for a class
#include <iostream>
using namespace std;
class Date
{
   int  da, mo, yr;
public:
   Date(int d, int m, int y)
   {
      da = d; mo = m; yr = y;
   }
   friend ostream& operator<<(ostream& os, const Date& dt);
};
ostream& operator<<(ostream& os, const Date& dt)
{
   os << dt.da << '/' << dt.mo << '/' << dt.yr;
   return os;
}

int main()
{
   Date dt(18, 2, 2023);
   cout << dt;
}
```

Output :

```
18/2/2023
```

Code to Celebrate
Celebrate to Code

Views of Success

Success is always doing your best.

Success is setting concrete goals.

Success is understanding the difference between need and want.

Success is believing you can.

Success is remembering to balance work with passion.

Success is knowing your life is filled with abundance.

Success is overcoming fear.

Success is learning something new each day.

Success is learning that losing a few battles can help you win a war.

Success is standing your ground when you believe in something.

Success is not giving up.

Success is never letting a disability hold you back.

Success is understanding that you control your destiny.

Success is celebrating small victories.

Success is loving and being loved back.

- **KR**