	SVKM's NMIMS Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Experiment: 7

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Aim: Programming using user defined functions & strings (character array)

Learning Outcomes: The learner would be able to

1. Understand the syntax of function declaration, definition and call.
2. Solve problems using user defined functions
3. write program using recursion
4. Use string handling functions
5. Implement programs using inline functions and macros

Theory:

Functions

- A function is a subprogram that performs a well-defined task.
- Every C++ program is a collection of functions.
- There should be at least one function called main( ).
- Every program execution starts with main( ) & main( ) is only function known by the Operating System, it is used as system call. Its default return type is int.
- C++ supports two types of functions
    - Built-in Library Functions
      C++ provides these functions.
      Examples are, sqrt(), pow(), sqrt(), strlen() etc.
    - User Define Functions
        - Programmer can define their functions.

- **Advantages of functions:**
    - Program maintenance & debugging is easy
    - Reusability of code.
    - Reduces program development time.
    - Work distribution is possible.
    - Easy to understand & easy to write.
    - Reduces coding size.

User Define Functions: -

- User can define their functions.
- Basic Parts/Components/Elements of functions: -
    1. Function Prototype Or Function Declaration.
    2. Function Calling or invocation
    3. Function Definition.

Function Prototype or Declaration.

- Like variables, functions are also declared called function prototype.
- Function should be declared before they are used.
- We can declare function inside main() or above the main().
- Function declaration or prototype is used to tell the compiler about following:
    - Return type of function.
    - Number of parameters or arguments.
    - Type of parameters.
    - Name of the function.
- **Syntax:-**

  return_type (datatype arg1, datatype arg2, ……..datatype argn);
  - In the above Syntax, we can also eliminate the names of arguments or skip function declarations in C++.
- **Example:-**
    - int add(int a, int b);
    - int add(int, int);

Function Calling.

- Function is called when a semi-comma follows a function name.
- Whenever a function is called, actual parameters of calling functions are copied into formal parameters of called function(if any).
- The number of actual and formal parameters should be the same(if any).
- The names of actual and formal parameters may be the same or different(if any).
- We can call functions using(also called Parameter Passing Techniques) call by value, call by reference & call by address.

- **Syntax:-**

  function_name(parameter_list);

- **Example:-**
    1. add(a,b);
    2. add(&a, &b);
    - Example 1 shows call by value & 2 is call by address.

Function Definition.

- User define functions definition consists of the following two parts.
- Function Header.
  - it contains "return type", "function name," & "argument list."
  - A semi-colon does not terminate the function header.
- Function Body
  - The function body should be enclosed in curly braces „{„ & „}".
  - It contains executable statements along with return value, if any.
  - These executable statements perform well-defined tasks.
- **Syntax:-**

```
return_type function_name ( argument_list){
        //function body.
}
```

- **Example:-**

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

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Write a program to find simple interest using user defined function...

Function Prototype/Declaration

Return Type of a function

Returning Value of SI

Function call / invocation

Function Definition

Formal Parameters or Formal Arguments

Actual Parameters or Actual Arguments

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

float simpleinterest(float, float, float);

float simpleinterest(float p, float t, float r){
   float SI;
   SI = (p*t*r)/100;
   return SI;
}

int main() {
   float p,t,r,si;
   cout<<"Enter amount time and rate of
            intrest"<<endl;
   cin>>p>>t>>r;

   si=simpleinterest(p,t,r);

   cout<<"Simple Interest is  "<<si;
   return 0;
}
```

Note: - Actual parameters and formal parameter may have same name but they are different

Output:-

Enter amount time and rate of interest

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Categories of Functions

WAP to perform addition of two numbers, without using function	WAP to perform addition of two numbers, using user defined function			
	1. Function without Argument & Without Return Type	**2. Function without Argument & with Return Type**	**3. Function with Argument & without Return Type**	**4. Function with Argument & with Return Type**

#include <iostream>
using namespace std;
 	
int main() {
int a,b;
cout<<"Enter two number"<<endl;
cin>>a>>b;
int c=a+b;
cout<<"Add is "<<c;
return 0;
}
	#include <iostream>
	using namespace std;
	
	void add(){
	int a,b;
	cout<<"Enter two number"<<endl;
	cin>>a>>b;
	int c=a+b;
	cout<<"Add is "<<c;
	}
	
	int main() {
	add();
	return 0;
	}
		#include <iostream>
		using namespace std;
		
		int add(){
		int a,b;
		cout<<"Enter two number"<<endl;
		cin>>a>>b;
		int c=a+b;
		return c;
		}
		
		int main() {
		int d = add();
		cout<<"Add is "<<d;
		return 0;
		}
			#include <iostream>
			using namespace std;
			
			void add(int a,int b){
			int c;
			c=a+b;
			cout<<"Add is "<<c;
			}
			
			int main() {
			int a,b;
			cout<<"Enter two number"<<endl;
			cin>>a>>b;
			add(a,b);
			return 0;
			}
				#include <iostream>
				using namespace std;
				
				int add(int a,int b){
				int c;
				c=a+b;
				return c;
				}
				
				int main() {
				int a,b;
				cout<<"Enter two number"<<endl;
				cin>>a>>b;
				int c=add(a,b);
				cout<<"Add is "<<c;
				return 0;
				}

Enter two number 3 5 Add is 8	Enter two number 3 5 Add is 8	Enter two number 3 5 Add is 8	Enter two number 3 5 Add is 8	Enter two number 3 5 Add is 8

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Parameter Passing Techniques

Call by Value vs call by reference vs call by address

Call By Value	Call By Reference	Call By Address
Normal/Ordinary Variables are used as a parameter	Reference Variables are used as a parameter	Pointer variables are used as a parameter
Formal parameter can't update the value of actual parameter (refer line no 4 of output)	Formal parameter update the value of actual parameter (refer line no 4 of output)	Formal parameter update the value of actual parameter (refer line no 4 of output)

#include <iostream>
using namespace std;
void swap(int a,int b){
cout<<"In swap() before swapping a="<<a<<" b="<<b<<endl;
int c=a;
a=b;
b=c;
cout<<"In swap() after swapping a="<<a<<" b="<<b<<endl;
}
int main() {
int a=10,b=20;
cout<<"In main() before swap() call a="<<a<<" b="<<b<<endl;
swap(a,b);
cout<<"In main() after swap() call a="<<a<<" b="<<b<<endl;
return 0;
}
Output:-
In main() before swap() call a=10 b=20
In swap() before swapping a=10 b=20
In swap() after swapping a=20 b=10
In main() after swap() call a=10 b=20
	#include <iostream>
	using namespace std;
	void swap(int &a,int &b){
	cout<<"In swap() before swapping a="<<a<<" b="<<b<<endl;
	int c=a;
	a=b;
	b=c;
	cout<<"In swap() after swapping a="<<a<<" b="<<b<<endl;
	}
	int main() {
	int a=10,b=20;
	cout<<"In main() before swap() call a="<<a<<" b="<<b<<endl;
	swap(a,b);
	cout<<"In main() after swap() call a="<<a<<" b="<<b<<endl;
	return 0;
	}
	
	Output:-
	In main() before swap() call a=10 b=20
	In swap() before swapping a=10 b=20
	In swap() after swapping a=20 b=10
	In main() after swap() call a=20 b=10
		#include <iostream>
		using namespace std;
		void swap(int *a,int *b){
		cout<<"In swap() before swapping a="<<*a<<" b="<<*b<<endl;
		int c=*a;
		*a=*b;
		*b=c;
		cout<<"In swap() after swapping a="<<*a<<" b="<<*b<<endl;
		}
		int main() {
		int a=10,b=20;
		cout<<"In main() before swap() call a="<<a<<" b="<<b<<endl;
		swap(&a,&b); //call by address
		cout<<"In main() after swap() call a="<<a<<" b="<<b<<endl;
		return 0;
		}
		Output:-
		In main() before swap() call a=10 b=20
		In swap() before swapping a=10 b=20
		In swap() after swapping a=20 b=10
		In main() after swap() call a=20 b=10

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Passing Arrays to functions

Finding Smallest element in 1D Array…using function

```
#include <iostream>
using namespace std;
int minelement(int a[],int n){
   int m=a[0];
   for(int i=0;i<n;i++){
      if(m>a[i]){
         m=a[i];
      }
   }
   return m;
}
int main(){
   int a[] ={10,9,5,3,23,34},n=6,m;
   m=minelement(a,n);
   cout<<"Smallest element is"<<m;
}
```

Finding Smallest element in 2D Array (matrix)…using function

```
#include <iostream>
using namespace std;
int minelement(int a[][3],int m,int n){
   int min=a[0][0];
   for(int i=0;i<m;i++){
      for(int j=0;j<n;j++)
      if(min>a[i][j]){
         min=a[i][j];
      }
   }
   return m;
}
int main(){
int a[][3]={10,9,1,30,23,34,12,35,67},n=3,m=3,min;
   min=minelement(a,m,n);
   cout<<"Smallest element is "<<min;
}
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Macro Function	Inline Function
- Macros starts with # Example: o File Inclusion ▪ #include<iostream> o Macros as constants ▪ #define PI 3.147 o Macros with arguments (Macro Functions) ▪ #define max(a,b) a>b?a:b	- Inline function is defined with inline keyword
- Macros functions are expanded as inline at pre-processing by pre-processor (before compliler)	- Inline functions are expanded at compile time by compiler.
- Speedup the execution of the program	- Compiler check the execution time & expanded it if required.

- Example:-
Macro Function
Macro body
`#include <iostream>`
`#define add(a,b) (a+b)`
`using namespace std;`
`int main(){`
`int a=20,b=30;`
Macro Call
`int c = add(a,b);`
`cout<<"Add is "<<c;`
`}`
	- Example:-
	Inline function
	`#include <iostream>`
	`using namespace std;`
	`inline void add(int a,int b){`
	`int c=a+b;`
	`cout<<"Add is "<<c;`
	`}`
	`int main(){`
	`int a=20,b=30;`
	Function call
	`add(a,b);`
	`}`

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Recursion

- Recursive function or recursion is a function that contains a call to itself.
- Works similar to divide and conquer.
- Recursion must contain one exit condition that can be satisfied, otherwise the recursive function
- will call itself repeatedly **until the runtime stack overflows.**
- Example: -

```
#include<stdio.h>
main( ){
    cout<<"I will stop\n";
    main( );
}
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

```
// Program to find factorial of a number using
recursion
#include<iostream>
using namespace std;
int fact(int n) {
        if(n == 1)
            return 1;

        return n * fact(n - 1);
}
int main() {
        int n,f;
        cout << "Enter a positive integer: ";
        cin >> n;
        f= fact(n);
        cout << n << "! = " <<f;
        return 0;
}
```

```
// Program to find sum of a series using recursion
#include<iostream>
using namespace std;
int sum(int n) {
        if(n == 1)
            return 1;

         return n + sum(n - 1);
}
int main() {
        int n,s;
        cout << "Enter a positive integer: ";
        cin >> n;
        s=sum(n);
        cout <<"Sum=" <<s;
        return 0;
}
```

```
// Program to find sum of following series using
function (Recursive function call)
1!+2!+3!+4!+5!+……+N!
#include<iostream>
using namespace std;
int fact(int n) {
        if(n == 1)
            return 1;
          return n * fact(n - 1);
}
int sum(int n) {
        if(n == 1)
            return 1;
          return fact(n) + sum(n - 1);
}
int main() {
  int n,s;
  cout << "Enter a positive integer: ";
  cin >> n;
  s= sum(n);
  cout <<"Sum is"<<s;
  return 0;
}
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Draw flowchart and write algorithm for printing the factorial of n using subroutine(function), where n is given as input.

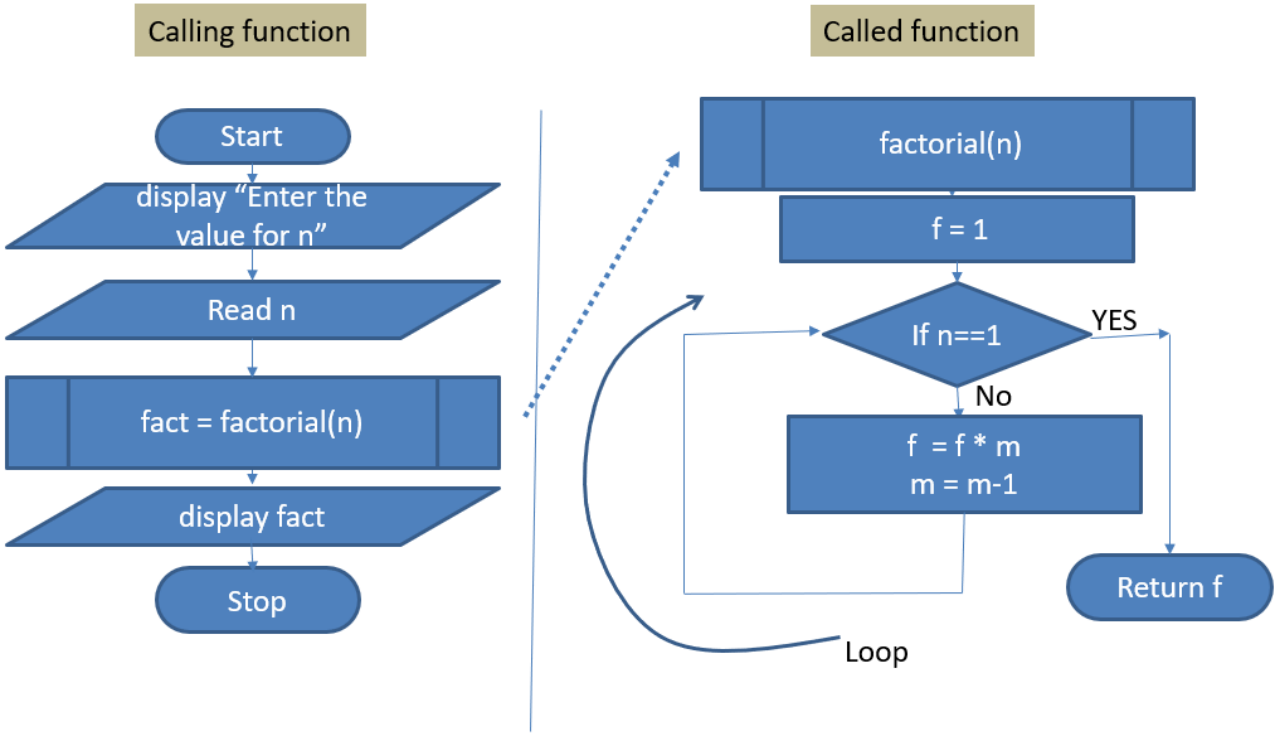
	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Algorithm :

1. Start
2. Display "Enter the value for n
3. Read n
4. fact = Factorial(n)
5. Display fact
6. Stop

Subroutine Factorial(n)

1. Start
2. f=1
3. If(n==1) Then
   Go to step 07
4. f = f *n
5. n = n-1
6. Goto Step 03
7. return f



Strings (Using Character Arrays)

- String is an array of characters
- Strings terminated with '\0'
- Declaration of String…

  Syntax:-

  ```
  char string_name[size];
  ```

  example:-

  ```
  char month[15];

  char studentname[20];

  char s1[10],s2[10];
  ```

- Initialization of String…

  Syntax:-

  ```
  char string_name[size] = {characters in single inverted comma, separated with comma};
  ```

  or

  ```
  char string_name[size] = "string";
  ```

  Example:-

  ```
  char subject[10] = {'c','p','p', '\0' };
  char name[10] = "cpp";
  char name[ ] = "cpp";
  ```

- Reading String…
  - There are various ways to read the stings, we can read it using **cin** statement

- Displaying String…
  - There are various ways to display stings, we can use **cout** statement to display

Write a program to initialize and display sting…

```
#include<iostream>

using namespace std;

int main(){

   char s[15]="Programming";

   cout<<"String is "<<s;
```

```
}
```

Write a program to read string from user and display it…

```
#include<iostream>
using namespace std;
int main(){
    char s[15];
    cout<< "Enter your name";
    cin>>s;
    cout<<"Welcome"<<s;
}
```

Write a program for Traversing/Processing a String Using Loop…

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s[15]="Programming";
    cout<<"String is \n";
    for (int i=0;s[i]!='\0';i++)
        cout<<s[i];
}
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

…String Handling Functions…

- There are various string handling functions available in **cstring** header file

  string **len**gth - strlen(s1)

  string **rev**erse - strrev(s1)

  string **l**o**w**e**r** - strlwr(s1)

  string **up**pe**r** - strupr(s1)

  string **c**o**py** - strcpy(s1,s2) & strncpy(s1,s2,n)

  string **c**onc**at**enation – strcat(s1,s2) & strncat(s1,s2,n)

  string **c**o**mp**arison – strcmp(s1,s2) & strncmp(s1,s2,n)

  string **ch**a**r**acter - strchr(s1,ch)

  string **r**everse **ch**a**r**acter - strrchr(s1,ch)

String Length…

- strlen(str_name);

WAP to find length of a string using string handling function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s1[10]="C++",s2[15]="Programming";
   int l1,l2;
   l1=strlen(s1);
   l2=strlen(s2);
   cout<<l1<<" "<<l2;
}
```

//WAP to Find length of a string without using string handling function…

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s[15]="Programming";
   int l=0;
   for (int i=0;s[i]!='\0';i++)
```

```
        l++;
    cout<<l;
}
```

String Concatenation…

- strncat(s1,s2)

WAP to concatenate two strings using string handling function

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="Kit",s2[10]="Kat";
    strcat(s1,s2);
    cout<<" String is " <<s1;
}
```

String Copy…

- strncpy(s1,s2)

WAP to copy on string to another using string handling function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="CPP",s2[10]="Python";
    strcpy(s1,s2);
    cout<<"After Copying String is "<<s1;
}
```

- strncpy(s1,s2,n)

WAP to copy n characters in one string to another using string handling function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="",s2[10]="CPP";
    strncpy(s1,s2,1);
```

```
    cout<<"After Copying String is "<<s1;
}
```

String Compare…

- strcmp(s1,s2)

WAP to compare two strings using string handling function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="CPP",s2[10]="CPP2";
    int c = strcmp(s1,s2);
    if(c==0)
        cout<<"Equal";
    else
        cout<<"Not Equal";
}
```

- strncmp(s1,s2,n)

WAP to compare two strings upto n characters using string function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="C++",s2[10]="CPP2";
    int c = strncmp(s1,s2,1);
    if(c==0)
        cout<<"Equal";
    else
        cout<<"Not Equal";
}
```

WAP to demonstrate strchr() function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
    char s1[10]="Dhananjay",ch='a';
    char *s2=strchr(s1,ch);
    cout<<s2;
```

```
}
```

WAP to demonstrate strrchr() function

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s1[10]="Dhananjay",ch='a';
   char *s2=strrchr(s1,ch);
   cout<<s2;
}
```

WAP to find reverse of a string using string function.

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s1[10]="madaM";
   strrev(s1);
   cout<<s1;
}
```

WAP to demonstrate working of strlwr()

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s[10]="SHALVI";
   strlwr(s);
   cout<<s;
   return 0;
}
```

WAP to demonstrate working of strlupr()

```
#include<iostream>
#include<cstring>
using namespace std;
int main(){
   char s[10]="Shalvi";
   strupr(s);
   cout<<s;
   return 0;
}
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**

Tasks:

Sr. No.	Problem Statement	I/O	Test Cases	Flow chart	Program- with color codes	Trace Table
1	Write a function that takes one integer parameter as a year, and displays its leap year or not.					
2	Write a function that takes two integer parameters x & y, and returns the result X^Y. (Don't use pow( ))					
3	Implement a program using user defined function to return largest of three floating-point numbers.					
4	WAP using user defined function to calculate and return factorial of a given integer.					
5	Write a menu driven program to compute sum of digits of a number, to find reverse of a number, to count number of digits by writing three different functions with parameters and return type.					
6	Write user defined function "search" to search element is present in 1D array or not. Search function accepts array and key to search as parameters.					
7	Write a program to print Fibonacci series up to n using recursion.					
8	Write one program to perform following operations on strings a) To find length of a string b) To compare two string for equality c) To Copy one string to other d) To concatenate two string e) To find reverse of a String					
9	WAP to copy one string to another string without using string handling function and display copied string.					

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**

B. Tech/MBA Tech	**Workbook**	**Academic Year- 2023-24**
Year:-First	**Subject:-** Programming for Problem Solving	**Semester: - First**