

NAME : _____
STUDENT NO. : _____
GROUP : _____

LESSON 1: PREDEFINED FUNCTIONS

Predefined or built-in functions are functions used by the programmers in order to speed up program writing. Programmers may use the existing code to perform common tasks without having to rewrite any code.

Header file	Commonly used function	Description
iostream.h	get()	For console input (keyboard) of type char
	put()	For console output (monitor) of type char
iomanip.h	setw(x)	Set field width to x
	setfill(x)	Set the fill character with x
	setprecision(x)	Set the floating-point precision to x
string.h	strcmp(s1, s2)	Compares one string to another
	strcpy(s1, s2)	Copies one string into another
	strlen(s)	Calculates the length of a string
	strcat(s1)	Appends one string to another
char.h	toupper(c)	Converts character c from lowercase to uppercase letter
	tolower(c)	Converts character c from uppercase to lowercase letter
	isdigit(c)	Returns true if c is a digit
	isalpha(c)	Returns true if c is an alphabet
math.h	pow(x, y)	Returns x raise to the power of y
	sqrt(x)	Square root of x
stdlib.h	abs(i)	Converts to the absolute value of i
	rand()	Generate a random positive integer between 0 and 32757

EXERCISE 1

Modify the given program so that the program is able to calculate and display the number of male and female employees.

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

int main()
{
    char name[30];
    char gender;

    for(int counter = 0; counter < 5; counter++)
    {
        cout<<"Name of the employee: ";
```

```
        cin>>ws;
        cin.getline(name, 30);
        cout<<"Gender of the employee (M / F): ";
        cin>>gender;
    }
    system("pause");
    return 0;
}
```

EXERCISE 2

Modify the codes given below so that the program is able to display the number of digits and alphabets entered by the user.

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

int main()
{
    char input;

    for(int counter = 0; counter < 10; counter++)
    {
        cout<<"Enter a symbol: ";
        cin>>input;
    }

    system("pause");
    return 0;
}
```

LESSON 2: PARAMETER PASSING BY REFERENCE

When there is *more than one value* to be returned by the calling function or changed during the program execution, reference parameter is used.

When a parameter is sent by reference, the function actually gets the *memory location* of the actual parameter it can directly access the data. In other words, the address of the variable in the computer's memory is actually being passed. As a result, the *changes made to the variable will affect the original*.

When passing by reference, the actual parameter in the calling function must be variable and the formal parameter must use the reference operator, which is referred by the symbol; &. The reference parameter forces the corresponding actual and formal parameters to refer to the same primary location.

In addition, the returned-type for passing parameters by reference is typically given the `void` type.

:: Function prototype ::

```
void getMinMax(int, int, int&, int&);
```

:: Function call ::

```
getMinMax(num1, num2, minimum, maximum);
```

:: Function definition ::

```
void getMinMax(int a, int b, int& min, int& max)
{
    if(a > b)
    {
        min = b;
        max = a;
    }
    else
    {
        min = a;
        max = b;
    }
}
```

Example of passing by reference.

EXERCISE 3

Compile and run the given complete program in C++. Discuss the output.

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

void displayValue(int&, int);

int main()
{
    int num = 40;
    int num2 = 45;
    cout<<"Value of num before function call = "<<num<<endl;
    cout<<"Value of num2 before function call = "<<num2<<endl;
    displayValue(num, num2);
    cout<<"\nValue of num after function call = "<<num<<endl;
    cout<<"Value of num2 after function call = "<<num2<<endl;

    cout<<endl;
    system("pause");
    return 0;
}

void displayValue(int& n, int m)
{
    cout<<"\nOriginal value of n = "<<n<<endl;
    cout<<"Original value of m = "<<m<<endl;

    n = n * 2;
    m = m * 3;

    cout<<"\nUpdated value of n = "<<n<<endl;
    cout<<"Updated value of m = "<<m<<endl;
}
```



EXERCISE 4

Using the function `findSquare()` that receives an input and return the square of the input, write a complete program that calculates and displays the squares of the integers from 1 to 50.

