Department of Computer Science & Engineering
University of Rajshahi

Object Oriented Programming with C++ Lab

Lab Manual

CSE1222

Prepared By,
Sujan Kumar Roy
Lecturer, Dept. of CSE, RU

University of Rajshahi
Rajshahi 6205, Bangladesh

C++ Class, Array, Object, Pointer & Reference

Assignment1

Title: Write a program that implements a simple C++ class named as myClass.

Outline: This assignment is a simple representation of C++ class declaration and implementation style.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare some member functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class myClass
{
int i;
public:
myClass(int n) { i=n;}
int get_i() { return i; }
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment2

Title: Write a program that implements a simple C++ class whose functions is to return the square roots of any positive number.

Outline: This assignment is a an example of C++ overloading

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax

- Declare member functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class sRoot
{
……..
public:
void SQRT(int n);
void SQRT(float n);
void SQRT(double n);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment3

Title: Write a program that implements a simple C++ class whose functions is to return the absolute value of any number.

Outline: This assignment is a an example of C++ overloading

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare member functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class absValue
{
……..
public:
void ABS(int n);
void ABS(float n);
void ABS(double n);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment4

Title: Write a program that implements a simple C++ class with constructor and destructor functions.

Outline: This assignment is an example of the use of Constructor and Destructor functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare Constructor and Destructor functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class myClass
{
……..
public:
myClass();
~myClass();
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment5

Title: Write a C++ program in which a Constructor function can take parameters.

Outline: This assignment is an example of the basic use of Constructor functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare Constructor and Destructor functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class myClass
{
Int n;
public:
myClass(int x) {x=n;}
~myClass();
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment6

Title: Write a C++ program that implements a STACK including push and pop operations.
Outline: This assignment is an example of the basic use of C++ class for STACK implementation.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare push and pop functions
- Run and Test the program including overflow and underflow test.

The following is an example of a simple C++ class.

```
class myStack
{
……
public:
void push(int n);
int pop();
```

};

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 50%.

Assignment7

Title: Write a C++ program that implements the use of an automatic inline function.
Outline: This assignment is an example of the basic use of inline function.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Declare inline functions
- Run and Test the program

The following is an example of a simple C++ class.

```
class samp
{
Int a, b;
public:
samp (int x, int y){a=x, b=y;}
int product() {return a*b;}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 50%.

Assignment8

Title: Write a C++ program that assigns one objects to another object of a class.
Outline: This assignment is an example of C++ object assignment.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Create class object
- Run and Test the program

The following is an example of a simple C++ class.

```
class myClass
{
Int a, b;
public:
myClass (int x, int y){a=x, b=y;}
void show() {cout<<a<<' '<<b;}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 20%.

Assignment9

Title: Write a C++ program that is capable of Passing Objects to Functions.
Outline: This assignment is an example of C++ object used as function argument.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function and create class object
- Run and Test the program

The following is an example of a simple C++ class.

```
class samp
    {
        int i;
        public:
            samp(int n) { i=n;}
            int get_i() { return i; }
    };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment10

Title: Write a C++ program that is capable of Returning Objects to Functions.
Outline: This assignment is an example of C++ object used as function return type.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function and create class object
- Run and Test the program

The following is an example of a simple C++ class.

```
class samp
    {
        int i;
        public:
            samp(int n) { i=n;}
            int get_i() { return i; }
    };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment11

Title: Write a C++ class that implements copy constructor.
Outline: This assignment is an example of copy constructor of C++.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function and create class object
- Run and Test the program

The following is an example of a simple C++ class.

```
class StringLength
    {
        public:
        char *p;
        void Print(void)
        {
        cout << "\f" << p << "\tLength is: "<< strlen(p)<<endl;
        }
        StringLength() {p = new char[50];}  // Constructor
        StringLength(StringLength &);   // Copy Constructor
    };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 60%.

Assignment12

Title: Write a C++ class that contains two classes car and track. The car class contains two private variables passengers and speed. The track class contains two private variables load and speed. Use friend function to compare the speed of car and track.

Outline: This assignment is an example of C++ friend function.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax for Car and Track
- Use member function
- Run and Test the program

The following is an example of a simple C++ class.

```
class truck;     //A Forward Declaration
class car
{
    int passengers;
    int speed;
    public:
            car(int p, int s) {passengers = p; speed = s; }
            friend int sp_greater(car c, truck t);
};
```

```
class truck
{
        int weight;
        int speed;
        public:
                        truck(int w, int s) {weight = w, speed = s;}
                        friend int sp_greater(car c, truck t);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment13

Title: Write a C++ program using a stack class, that contains a function called loadstack () that returns a stack that is already loaded with the letters of the alphabet (a-z). Assign this stack to another object in the calling routine and prove that it contains the alphabet. Be sure to change the stack size so it is large enough to hold the alphabet.

Outline: This assignment is an example of C++ stack class.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function
- Run and Test the program

The following is an example of a simple C++ class.

```
class loadStack
        {
                char q[SIZE];
                int rare;
                int front;
                public:
                loadStack() {rare = 0;front=0;}
                void push(char ch);
                char pop();
        };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program

- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment14

Title: Write a C++ program that creates a one/ two dimensional array of object and initialized them.
Outline: This assignment is an example of object used as objects.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function
- Run and Test the program

The following is an example of a simple C++ class.

```
//One Dimensional Array of Object Example
        samp ob[4] = {-1, -2, -3, -4};
        for (i=0; i<4; i++)
        cout<< ob[i].get_a() << ' ';
        cout << "\n";
//Two Dimensional Array of Object Example
                samp ob[4][2] = {
                        1, 2,
                        3, 4,
                        5, 6,
                        7, 8
                  };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment15

Title: Write a C++ program that can allocate objects dynamically.
Outline: This assignment is an example of dynamic object allocation.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function
- Run and Test the program

The following is an example of a simple C++ class (use previous samp class).

```
int samp *p;
p= new samp;
if (!p) {
        cout << "Allocation error\n";
        return 1;
       }
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment16

Title: Write a C++ program that can pass and return reference to a function.
Outline: This assignment is an example of pointer and reference.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use member function
- Run and Test the program

The following is an example of a simple C++ class (use previous samp class).

```
class myclass {
        int who;
        public:
        myclass(int n){
        who= n;
        cout<< "constructing " << who << "\n";
                        }
        ~myclass () { cout << "Destructing" << who << "\n";}
        int id () {return who;}
       };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives

- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 50%.

Function Overloading

Assignment17

Title: Write a C++ program that can overload a constructor function.
Outline: This assignment is an example of function overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use constructor function
- Run and Test the program

The following is an example of a simple C++ class

```
class myclass
{
        int x;
        public:
        myclass() {x=0; }
        myclass(int n) {x=n;}
        int getx() {return x;}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 50%.

Assignment18

Title: Write a C++ program that implements a copy constructor function and allows string type objects.
Outline: This assignment is an example of copy constructor.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use copy constructor function syntax
- Run and Test the program

The following is an example of a simple C++ class

```
class strtype
 {
        char *p;
        public:
        strtype(char *s);
        strtype(const strtype &o);
        ~strtype() {delete [] p;}
        char *get() { return p;}
 };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment19

Title: Write a C++ program that allows a function to use default argument.
Outline: This assignment is an example of default argument.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use function syntax
- Run and Test the program

The following is an example of a simple C++ class

```
class myclass
{
        int x;
        public:
        myclass(int n = 0) {x = n; }
        int getx() {return x; }
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 40%.

Assignment20

Title: Write a C++ program that allows a function to use default argument.

Outline: This assignment is an example of default argument.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use function syntax
- Run and Test the program

The following is an example of a simple C++ class

```
class myclass
{
        int x;
        public:
        myclass(int n = 0) {x = n; }
        int getx() {return x; }
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.

Percentage of Weight: 40%.

Assignment21

Title: Write a C++ program that function overloading ambiguity.

Outline: This assignment is an example of function overloading ambiguity.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use function syntax
- Run and Test the program

The following is an example of a simple C++ class

```
int f (int a)
{
        Return a*a;
}
int f (int a, int b=0)
{
        Return a*b;
}
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 40%.

Assignment22

Title: Write a C++ program that overloads a function three times.
Outline: This assignment is an example of function overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use function syntax
- Run and Test the program

The following is an example of a simple C++ class

```
void f(int i); // integer parameter
void f(int i, int j); // two integer parameters
void f(double k); // one double parameter
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 40%.

Operator Overloading

Assignment23

Title: Write a C++ program that create a class that is relative to coordinate class of two variables (x, y) and overload the +, -, * and / operators.
Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   double x,y;
   public:
   coord() {x=0;y=0;}
   coord(int i,int j){x=i; y=j;}
   void get_xy(int &i, int &j){i=x; j=y;}
   coord operator+(coord ob2);
   coord operator-(coord ob2);
   coord operator*(coord ob2);
   coord operator/(coord ob2);
 };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment24

Title: Write a C++ program that create a class that is relative to coordinate class of two variables (x, y) and overload the -- operator.

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   int x,y;
   public:
   coord(){x=0;y=0;}
   coord(int i, int j){x=i;y=j;}
   void get_xy(int &i, int &j){i=x;j=y;}
   coord operator--();
   coord operator--(int post);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program

- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment25

Title: Write a C++ program that create a class that is relative to coordinate class of two variables (x, y) and overload the + operator so that it is both a binary operator and a unary operator. When it is used as a unary operator, have the + make any negative coordinate value positive.

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   int x,y;
   public:
   coord() {x=0;y=0;}
   coord(int i,int j){x=i; y=j;}
   void get_xy(int &i, int &j){i=x; j=y;}
   coord operator+(coord ob2);
   coord operator+();
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment26

Title: Write a C++ program that create a class that is relative to coordinate class of two variables (x, y) and use the friend operator functions for overloading the + operator .

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   int x,y;
   public:
   coord(){x=0;y=0;}
   coord(int i, int j){x=i;y=j;}
   void get_xy(int &i, int &j){i=x; j=y;}
   friend coord operator+(coord ob,int i);
   friend coord operator+(int i,coord ob);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.

Percentage of Weight: 80%.

Assignment27

Title: Write a C++ program that create a class that is relative to coordinate class of two variables (x, y) and use the friend operator functions for overloading the ++ operator .

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   int x,y;
   public:
   coord(){x=0;y=0;}
   coord(int i, int j){x=i;y=j;}
   void get_xy(int &i, int &j){i=x; j=y;}
   friend coord operator++(coord &ob);

};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment28

Title: Write a C++ program that creates two classes that is relative to coordinate class of two variables (x, y) and use the friend operator functions to add the two coordinates.
Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord1;

class coord2
{
   int x,y;
   public:
   coord2(){x=0;y=0;}
   coord2(int i, int j){x=i;y=j;}
   friend coord1 operator+(coord1 ob1,coord2 ob2);
};

class coord1
{
   int x,y;
   public:
   coord1(){x=0;y=0;}
   coord1(int i,int j){x=i;y=j;}
   void get_xy(int &i,int &j){i=x;j=y;}
   friend coord1 operator+(coord1 ob1,coord2 ob2);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm

- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment29

Title: Write a C++ program that creates a class that is relative to coordinate class of two variables (x, y) and overload the << and >> shift operators so that the following types of operations are allowed: ob<<integer and ob>>integer. Make sure the operators shift the x and y values by the amount specified.

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class coord
{
   int x,y;
   public:
   coord(){x=0;y=0;}
   coord(int i, int j){x=i;y=j;}
   void get_xy(int &i,int &j){i=x; j=y;}
   coord operator<<(int i);
   coord operator>>(int i);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment30

Title: Write a C++ program that creates a class that can overload the &&, ==, ! = and || operators.

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class three_d
{
   int x,y,z;
   public:
   three_d(int i, int j, int k)
   {
      x=i; y=j; z=k;
   }
   three_d(){x=0;y=0;z=0;}
   void get(int &i, int &j, int &k)
   {
      i=x; j=y; k=z;
   }
   int operator&&(three_d ob2);
   int operator==(three_d ob2);
   int operator!=(three_d ob2);
   int operator||(three_d ob2);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment31

Title: Write a C++ program that creates a String Type class which allows the following types of operations:
* String concatenation using the +operator
* String assignment using the = operator
* String comparisons using <, >, and == operators

Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class strtype
{
   char s[80];
   public:
   strtype(){*s='\0';}
   strtype(char *p){strcpy(s,p);}
   char *get(){return s;}
   strtype operator+(strtype s2);
   strtype operator=(strtype s2);
   int operator<(strtype s2);
   int operator>(strtype s2);
   int operator==(strtype s2);
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment32

Title: Write a C++ program that creates a class that can overload the subscript [] operator.
Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class array
{
        int a[s];
        public:
        array();
        int operator[](int i)
        {
        return a[i];
        }
 };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment33

Title: Write a C++ program that creates a class that overloads the * operator to find the factorial of a big number greater than long integer data type.
Outline: This assignment is an example of operator overloading.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use operator overloading member functions
- Run and Test the program

The following is an example of a simple C++ class

```
class BigInt
{
public:
	char num[2000];
	int l;
	BigInt()
	{
		l=1;
		strcpy(num,"1");
	}
	BigInt operator*(int n);
	void show()
	{
		strrev(num);
		cout<<num<<endl;
		strrev(num);
	}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Inheritance, Virtual Functions & Template

Assignment34

Title: Write a C++ program in which a derived class inherits a base class as public.
Outline: This assignment is an example of a simple inheritance.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

The following is an example of a simple C++ class

```
class base {
        int x;
        public:
        void setx(int n) {x=n;}
        void showx() { cout << x << '\n'; }
        };
        class derived : public base {
        int y;
        public:
        void sety(int n) {y=n;}
        void showy() {cout << y << '\n'; }
           };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment35

Title: Write a C++ program in which a derived class inherits a base class as private.
Outline: This assignment is an example of a simple inheritance.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process

- Run and Test the program

The following is an example of a simple C++ class

```
class base {
        int x;
        public:
        void setx(int n) {x=n;}
        void showx() { cout << x << '\n'; }
        };
        class derived : private base{
        int y;
        public:
        void sety(int n) {y=n;}
        void showy() {cout << y << '\n'; }
           };
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 30%.

Assignment36

Title: Write a C++ program by using the following skeleton and create appropriate car() and truck() constructor function. Have each pass along appropriate arguments to vehicle. In addition have car() initialize passengers as specified when an object is created Have truck() initialize loadlimit as specified when an object is created.

Outline: This assignment is an example of base class and derived class inheritance process.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

Use the following skeleton to create the constructor

```
        class vehicle {
        int num_wheels;
        int range;
        public:
        vehicle(int w, int r)
        {
        num_wheels = w; range = r;
        }
        void show()
```

```
{
cout << "Wheels: " << num_wheels << '\n';
cout << "range:" << range << '\n';
}
};
class car : public vehicle {
int passengers;
public:
void show()
{
show();
cout << "passengers:" << passengers << '\n';
}
};
class truck : public vehicle {
int loadlimit;
public:
void show()
{
show();
cout << "loadlimit:" << loadlimit << '\n';
}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 70%.

Assignment37

Title: Write a C++ program which creates a virtual base class to prevent two copies of base from being present in derived3.

Outline: This assignment is an example of virtual base class.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

The following is an example of a simple C++ class

```
class base {
public:
```

```
int i;
};
class derived1 : virtual public base {
public :
int j;
};
class derived2 : virtual public base {
int k;
};
class derived3 : public derived1, derived2 {
public:
int product() { return i * j * k; }
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 60%.

Assignment38

Title: Write a C++ program that creates a generic class building which contains number of floors, number of rooms and its total square footage. Create a derived class called house that inherits building class and contains number of bedrooms and bathrooms. Finally, create another class office that also inherits building class and contains the number of fire extinguisher and telephone.

Outline: This assignment is an example of inheritance.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

The following is an example of a simple C++ class

```
class building
{
Protected:
int floors;
int rooms;
double footage;
}

class house : public building
```

```
{
int bedrooms;
int bathrooms;
public:
house (use the desired arguments);
void show();
};

class office : public building
{
int phones;
int extinguisher;
public:
office(use the desired arguments);
void show();
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment39

Title: Write a C++ program that creates a base class called num. Have this class hold an integer value and contain a virtual function called shownum(). Create two derived classes called outhex and outoct that inherit num. Have the derived classes override shownnum() so that it displays the value in hexadecimal and octal respectively.

Outline: This assignment is an example of inheritance and virtual functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

The following is an example of a simple C++ class

```
class num{
        public:
        int i;
        num (int x) {i=x;}
        virtual void shownum()
```

```
    {cout<< i<< '\n';}};
class outhex : public num { public:
outhex(int n) : num (n)  {}
void showdown () {cout << hex<< i << '\n';}
};

class outoct : public num { public:
outoct(int n) : num (n)  {}
void showdown () {cout << oct<< i << '\n';}
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment40

Title: Write a C++ program that creates a base class called dist that stores the distance between two points in a double variable. In dist create a virtual function called trav_time() that outputs the time it takes to travel that distance assuming that the distance is in miles and speed is 60 miles per hour. In a derived class called metric override trav_time() so that it outputs the travel time assuming that the distance is in kilometers and the speed is 100 kilometers per hour.

Outline: This assignment is an example of inheritance and virtual functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Inheritance Process
- Run and Test the program

The following is an example of a simple C++ class

```
class dist
{
public:
double d;
dist(double f) {d=f;}
virtual void trav_time()
{
cout<<"Travel time at 60 mph:";
cout<<d/60<<'\n';
 }
};
```

```
class metric:public dist
{
public:
metric(double f): dist(f){}
void trav_time()

{
cout<<"Travel time at 100 mph:";
cout<<d/100<<'\n';
 }
};
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 70%.

Assignment41

Title: Write a generic function called min() that returns the lesser of its two arguments. For example min(3,4) will return 3, min(a,b) will return a and so on. Demonstrate your function in a program.

Outline: This assignment is an example of generic class functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Generic Function Declaration scheme
- Run and Test the program

The following is an example of a simple C++ class

```
template<class X > X min(X a, X b)
{
        if(a<=b)  return a;
        else return b;
}
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 70%.

Assignment42

Title: Write a C++ program that creates a temple function called find() which searches an array for an object and returns either the index of the matching object or -1 if no match is found. Here is the prototype for a specific version of find(). Convert find() into a generic function and demonstrate your solution within a program.

Outline: This assignment is an example of generic functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Generic Function Declaration scheme
- Run and Test the program

The following is an example of a simple C++ class

```
template <class X> int find(X object, X *list, int size)
{
  int i;
  for(i=0; i<size; i++)
        if(object == list[i]) return i;
        return -1;
}
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment43

Title: Write a C++ program that creates a very simple generic singly linked list class. It then demonstrates the class by creating a linked list that stores characters.

Outline: This assignment is an example of generic functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax

- Use Generic Function Declaration scheme
- Run and Test the program

The following is an example of a simple C++ class

```
template <class data_t>class list{
 data_t data;
 list *next;
 public:
 list(data_t d);
 void add(list *node){node->next=this; next=0;}
 list *getnext(){return next;}
 data_t getdata(){return data;}
 };
 template<class data_t>list<data_t>::list(data_t d)
 {
 data=d;
 next=0;}
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 90%.

Assignment44

Title: Write a C++ program that creates a generic bubble sort algorithm.

Outline: This assignment is an example of generic functions.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use Generic Function Declaration scheme
- Run and Test the program

The following is an example of a simple C++ class

```
template <class X> void bubble(X *data, int size)
{
  register int a, b;
  X t;
```

---apply bubble short algorithm---
}

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment45

Title: Write a program that demonstrates simple C++ exception handling process.

Outline: This assignment is an example of exception handling.

Task: In order to write the program, you should follow the following instructions.

- Use Class declaration syntax
- Use exception handling scheme
- Run and Test the program

The following is an example of a simple C++ class

Use try/catch and throw and generates examples to handle exceptions.

```
Try
{
        throw 10;
}
catch (int i)
{
        Cout<<"Caught i:"<<i;
}
```

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 60%.

Some Advance C++ Programs

Assignment46

Title: Create a C++ class called Rectangle with attributes length and width, each of which defaults to 1 and calculate the perimeter and the area of the rectangle.

Task to Implement: In order to write the program, you should follow the following instructions.
Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 80%.

Assignment47

Title: Write a program that creates a C++ class named SavingsAccount that is capable of storing different interests of a user.

Task to Implement: In order to write the program, you should follow the following instructions.
Use a static data member annualInterestRate to store the annual interest rate for each of the savers. Each member of the class contains a private data member savingsBalance indicating the amount the saver currently has on deposit. Provide member function calculateMonthlyInterest that calculates the monthly interest by multiplying the balance by annualInterestRate divided by 12; this interest should be added to savingsBalance. Provide a static member function modifyInterestRate that sets the static annualInterestRate to a new value. Write a driver program to test class SavingsAccount. Instantiate two different objects of class SavingsAccount, saver1 and saver2, with balances of TK2000.00 and TK3000.00, respectively. Set the annualInterestRate to 3 percent. Then calculate the monthly interest and print the new balances for each of the savers. Then set the annualInterestRate to 4 percent, calculate the next month's interest and print the new balances for each of the savers.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives

- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment48

Title: Write a C++ program to create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store.

Task to Implement: In order to write the program, you should follow the following instructions.
The Invoice should include four pieces of information as data members a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

Report Writing: For report writing instructions, just follow the following instructions:
- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment49

Title: Create a class called Employee that includes three pieces of information as data membersa first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. If the monthly salary is not positive, set it to 0.

Task to Implement: In order to write the program, you should follow the following instructions.
Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment50

Title: Create a class called number. The user should be able to enter input in any of the four number systems:

(i) Decimal (ii) Binary (iii) Octal (iv) Hexadecimal

Task to Implement: In order to write the program, you should follow the following instructions.
Create a constructor function that converts any given user input into decimal number first, and then use function overloading to convert this decimal input to its corresponding number systems.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment51

Title: Implement a person class. Each object of this class will represent a human being. Data members should include the person's name, year of birth. Include a default constructor, a destructor, access function and a print function.

Task to Implement: Use the appropriate methods and functions for the implementation of the program.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment52

Title: Implement a student class using the person class (Asssignment51). Include data members for student identification number, major program, grade point average, and credit earned. Include a member function update(int credit,char grade) that process the given information(credit and grade) for one course, using it to update the students grade point average and credits earned.

Task to Implement: Use the appropriate methods and functions for the implementation of the program.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

Assignment53

Title: Define a class named Sequence for storing sorted strings. Define a constructor, a destructor, and the following member functions for Sequence:

i. Insert which inserts a new string into its sort position.
ii. Delete which deletes an existing string.
iii. Find which searches the sequence for a given string and returns true if it finds it, and false otherwise.
iv. Print which prints the sequence strings.

Task to Implement: Use the appropriate methods and functions for the implementation of the program.

Report Writing: For report writing instructions, just follow the following instructions:

- Name of the Program
- Program Areas
- Objectives
- Algorithm
- Methods & Functions to be used

Mode of Evaluation: Presentation, Implementation & Testing, and Viva.
Percentage of Weight: 100%.

For the next assignments, the Report Writing, Mode of Evaluation, Percentage of Weight will be same as used in Assignment53.

Assignment54

(Date Class) Create a class called Date that includes three pieces of information as data members month (type int), a day (type int) and a year (type int). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 112; if it is not, set the month to 1. Provide a set and a get function for each data member. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class Date's capabilities.

Assignment55

Implement a **date** class with member data for the month, day and year. Each object of this class will represent a specific A.D. data, storing the month, day and year as integers. Include a default constructor, a copy constructor, access functions, a function **reset(int y,int m,int d)** to reset the date for an existing object, a function **advance(int y,int m,int d)** to advance an existing date by y years, m months and d days, and a **print()** function. Use a **normalize()** utility function to ensure that the data members are in the correct range.
1≤**year**
1≤**month**≤12
1≤**day**≤**daysIn(month)**
Where **daysIn(int month**) is another utility function that returns the number of days in month. Ignore leap years.

Assignment56

String concatenation requires two operands the two strings that are to be concatenated. In the text, we showed how to implement an overloaded concatenation operator that concatenates the second String object to the right of the first String object, thus modifying the first String object. In some applications, it is desirable to produce a concatenated String object without modifying the String arguments. Implement operator+ to allow operations such as string1 = string2 + string3;

Assignment57

Create a class RationalNumber (fractions) with the following capabilities: Create a constructor that prevents a 0 denominator in a fraction, reduces or simplifies fractions that are not in reduced form and avoids negative denominators. Overload the addition, subtraction, multiplication and division operators for this class. Overload the relational and equality operators for this class.

Assignment58

Create a class called Rational for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private data of the class the numerator and the denominator. Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided and should store the fraction in reduced form. For example, the fraction would be stored in the object as 1 in the numerator and 2 in the denominator. Provide public member functions that perform each of the following tasks:

a. Adding two Rational numbers. The result should be stored in reduced form.
b. Subtracting two Rational numbers. The result should be stored in reduced form.
c. Multiplying two Rational numbers. The result should be stored in reduced form.
d. Dividing two Rational numbers. The result should be stored in reduced form.
e. Printing Rational numbers in the form a/b, where a is the numerator and b is the denominator.
f. Printing Rational numbers in floating-point format.

Assignment59

Define a class named Complex for representing complex numbers. A complex number has the general form a + *i*b, where a is the real part and b is the imaginary part (*i* stands for imaginary). Complex arithmetic rules are as follows:

$$(a + ib) + (c + id) = (a + c) + i(b + d)$$
$$(a + ib) - (c + id) = (a + c) - i(b + d)$$
$$(a + ib) * (c + id) = (ac - bd) + i(bc + ad)$$

Define these operations as member functions of Complex.

Assignment60

Overload the following two operators for the Binary class:

Operator - which gives the difference of two binary values? For simplicity, assume that the first operand is always greater than the second operand.

Operator [ ] which indexes a bit by its position and returns its value as a 0 or 1 integer.

Assignment61

Overload the following two operators for the Binary class:

- Operator - which gives the difference of two binary values? For simplicity, assume that the first operand is always greater than the second operand.
- Operator [ ] which indexes a bit by its position and returns its value as a 0 or 1 integer.

Assignment62

Write overloaded versions of a Max function which compares two integers, two reals, or two strings, and returns the 'larger' one.

Assignment63

Create a base class called **shape**. Use this class to store two **double** type values that could be used to compute the area of figures. Derive two specific classes called **triangle** and **rectangle** from the base shape. Add to the base class, a member function to initialize base class data members and another member function, to compute and display the area of figures, which would be made as virtual function and redefine this function in the derived program that will accept dimensions of a triangle or rectangle interactively and display the area.

Assignment64

Write a program to read a text file. Store each word in a map. The key value of the map is the count of the number of times the word appears in the text. Define a word exclusion set containing words such as a, an, or, the, and, and but. Before entering a word in the map, make sure it is not present in the word exclusion set. Display the list of words and their associated count when the reading of the text is complete. As an extension, before displaying the text, allow the user to query the text for the presence of a word.

Assignment65

Implement a Quaternion class for hypercomplex numbers (also called "Hamiltonians"). Each object of this class will represent a hypercomplex number t+xi+yj+zk ; where each of the components t,x,y,z has type double.

Use the following class declaration syntax;

```
class Quaternion{
  double t,x,y,z;
  public:
   Quaternion(double i=0,double j=0,double k=0,double l=0){t=i;x=j;y=k;z=l;}
   Quaternion(const Quaternion &qq){  t=qq.t;   x=qq.x;   y=qq.y;  z=qq.z;  }
   inline void show();
   Quaternion operator+(Quaternion o1);
```

```
Quaternion operator-(Quaternion o1);
Quaternion operator*(Quaternion o1);
void access();
void normal();
};
```

Some Graphics Programs for C++

Assignment66

Write a C++ class named Box that represents a Box. Use a member function called initialize() for setting the initial position of the Box in the screen. Then use another two member function left() and right() whose function is to move the Box either left or right as left and right cursor key is pressed by the user.

Use the following class declaration syntax;

```
class graph
  {
 public:
    int xmax,xmin;
    graph();
    void initial(int n);
    void initial();
    void function();
    void right();
    void left();
  };
```

Assignment67

Write a C++ program that implements a dial watch. Your program should be capable of reading the current time from your system and start the watch from that time.

Use the following class declaration syntax;

```
class clock{     public:
                    clock();
                    void display();
              };
```

Assignment68

Write a C++ class puzzle that implements a simple number puzzle game. Use a constructor function called puzzle() for setting the initial position of the Puzzle in the

screen. Then use another four member function left() , right(), up() and down() whose function is to move the Box either left , right, up or down as left, right, up and down cursor keys are pressed by the user. Also use a member function named reset() that is capable of resetting the puzzle number positions.

Use the following class declaration syntax;

```
class puzzle
{
public:
  int a[16];
  puzzle(){   reset(); }
  void up();
  void down();
  void left();
  void right();
  void reset();
};
```