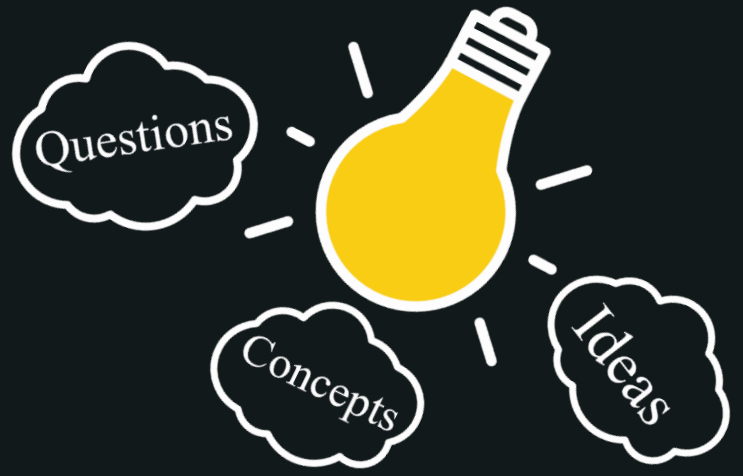


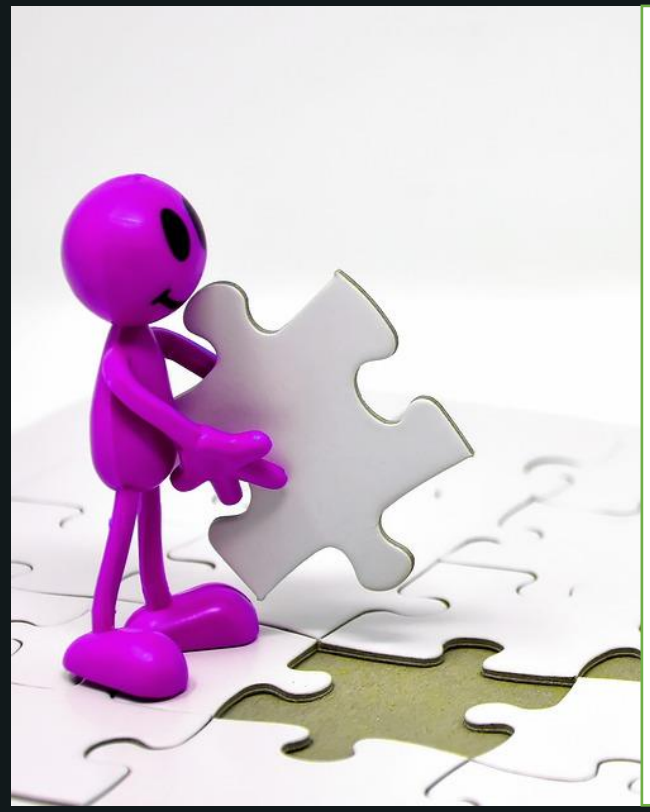
Friend Function & Class

C++ Programming

VEDINESH

Friend

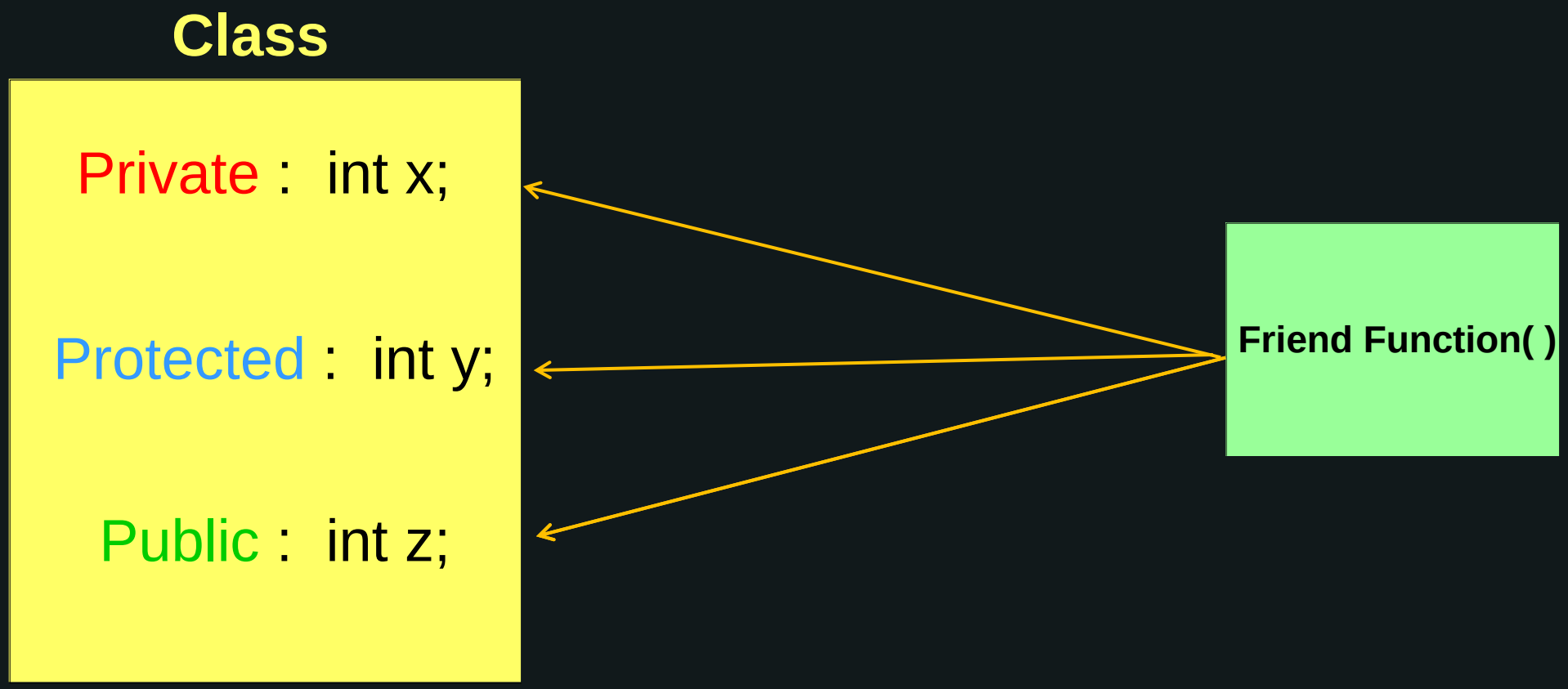
Friend



- Reduce Work Load
- Speed Up Work
- Less Chances Of mistake
- High Standard of output
- Better Work Management

Why Friend Function

VEDINESH



Friend Function

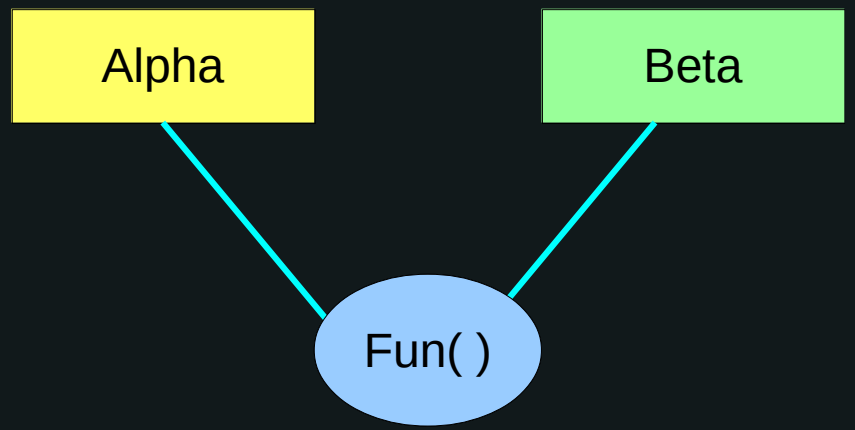
VEDINESH

```
class Alpha
{
  private:
            int a1;
  public:
            Alpha( int arg = 0)
            {  a1 = arg; }
            friend void Fun( );
};
```

```
class Beta
{
  private:
            int b1;
  public:
            Beta( int arg = 0)
            {  b1 = arg; }
            friend void Fun( );
};
```

```
 void Fun( )
 {
    Alpha a(8);
    Beta b(2);

    int x = a.a1 +  b.b1;

    cout << "Sum "  << x;
}
```

```
void main( )
{
   Fun( );
}
```



NOTE:- Friend function connecting two class or more(act as bridge).

Questions

VEDINESH

Class can have more than one friend function ? Yes

Can a same function can become friend of multiple classes ? Yes

Friend function can be invoked as normal function ? Yes

Do we need to create object of class in order to access friend function ? No

Friend Class

VEDINESH

Class Suzi

Class Jena

Class Suzi

Class Jena

Can Access and use features and functionalities of each other.

Friend Class

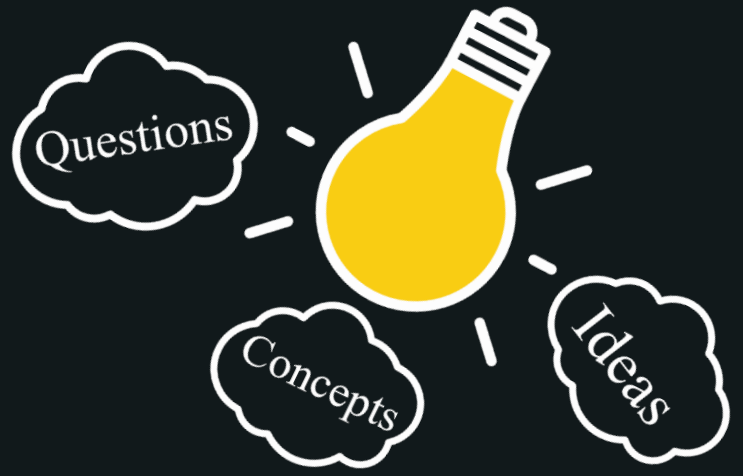
VEDINESH

```
class Alpha
{
 private:
            int a1;
 public:
            Alpha( int arg = 0)
            {  a1 = arg; }

            friend class Beta;
};
```

```
class Beta
{
 private:
            int b1;
 public:
            Beta( int arg = 0)
            {  b1 = arg; }

            void Sum( )
            {
              Alpha alpha_obj (3);

              int sum = alpha_obj.a1 + b1;

              return sum;
            }
};
```

```
 void main( )
 {
    Beta beta_obj(7);

    beta_obj.Sum( );
}
```

NOTE:-Now all memeber functions of Beta class can access private data of Alpha .



Static Member &Function

C++ Programming

VEDINESH

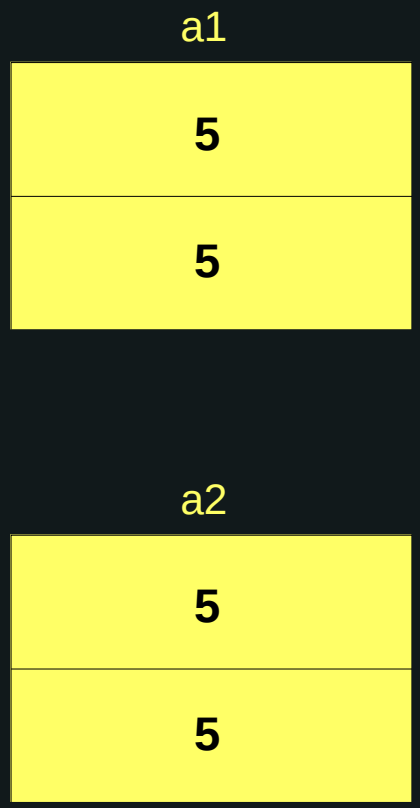
Static Members

VEDINESH

```
class Alpha
{
  private:
            int a;
            int b;
  public:
            Alpha( )
            {
              a = 5;
              b = 5;
            }
};
```

```
 void main( )
 {
    Alpha a1;
    Alpha a2;
}
```

NOTE:- Each Object will create separate copy of itself in memory .



Static Member

VEDINESH

```
class Alpha
{
 private:
            int a;
            int b;
 public:
            Alpha( )
            {
              a = 5;
              b = 5;
              stat ++;
            }

            static int stat;
};
```

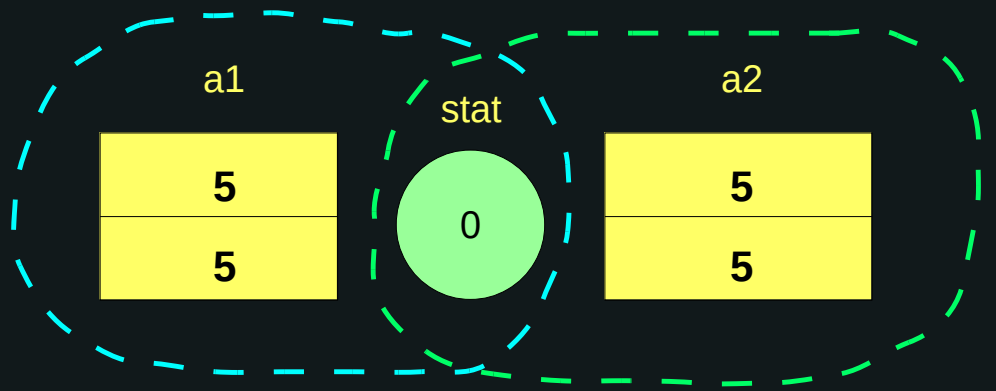
```
int  Alpha ::  stat = 0;
```

```
 void main( )
 {
    Alpha a1;
    Alpha a2;

    cout <<  a1.stat;

    cout <<  a2.stat;

    cout <<  Alpha::stat;
}
```

NOTE:- Static member would be allocated memory only once .

NOTE:- And that memory is shared by both the objects.

NOTE:- Static Data members belong a class & common to all objects.



Static Member Function

```
class Alpha
{ private:
            int a;
            int b;
  public:
            Alpha( )
            {  a = 5;
               b = 5;
            }

            static int stat;

            static int getStat( )
            {  stat ++ ;
               return stat ; }
};

int  Alpha ::  stat = 0;
```

```
void main( )
{
    cout << Alpha :: getStat( );
    Alpha a1;
    Alpha a2;
    cout << Alpha :: getStat( );
    cout << a1.getStat( );
    cout << a2.getStat( );
 }
```

NOTE:- Static member can only access static members .