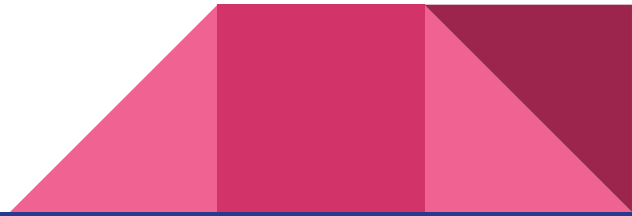
Object Oriented Programming

Topics covered

- Constructor overloading
- Function overloading
- Static data member
- Static member function
- Constant member function
- Inline function
- This pointer



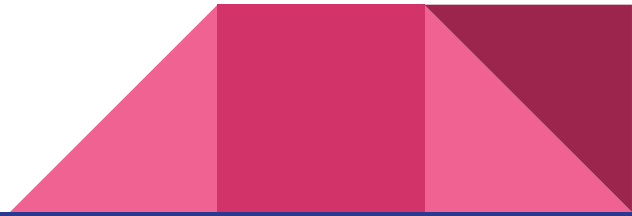
Constructor Overloading

In C++, We can have more than one constructor in a class with same name, as long as each has a different list of arguments.

- Overloaded constructors essentially have the same name (name of the class) and different number of arguments.
- A constructor is called depending upon the number and type of arguments passed.
- While creating the object, arguments must be passed to let compiler know, which constructor needs to be called.

Function overloading

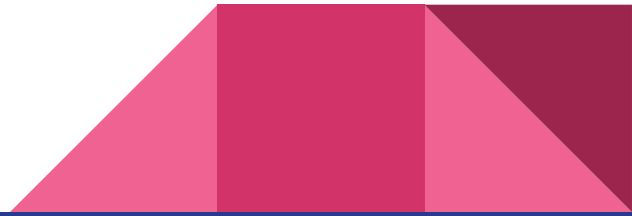
- Function overloading is a feature in C++ where two or more functions can have the same name but different parameters.
- Function overloading can be considered as an example of polymorphism feature in C++.
- Function overriding is a feature of OOPs Programming that allows us to override a function of parent class in child class.



Static data members

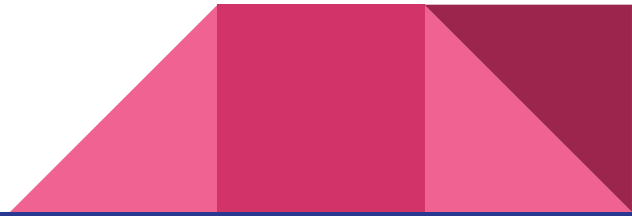
- Static data members are class members that are declared using the static keyword. There is only one copy of the static data member in the class, even if there are many class objects. This is because all the objects share the static data member. The static data member is always initialized to zero when the first class object is created.
- The syntax of the static data members

```
static data_type data_member_name;
```



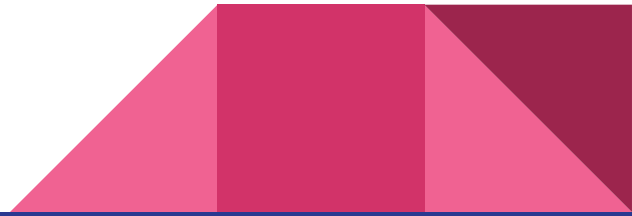
Static member function

- By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::.
- A static member function can only access static data member, other static member functions and any other functions from outside the class.
- Static member functions have a class scope and they do not have access to the this pointer of the class. You could use a static member function to determine whether some objects of the class have been created or not.



Const member function

- Like member functions and member function arguments, the objects of a class can also be declared as **const**. an object declared as const cannot be modified and hence, can invoke only const member functions as these functions ensure not to modify the object.
- A function becomes const when the const keyword is used in the function's declaration. The idea of const functions is not to allow them to modify the object on which they are called. It is recommended the practice to make as many functions const as possible so that accidental changes to objects are avoided.



Inline Function

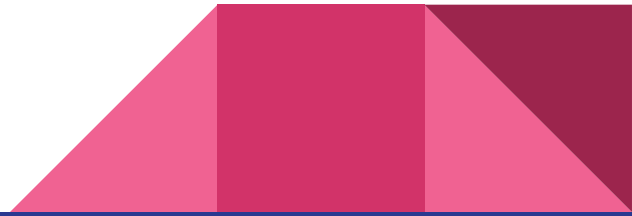
- If a function is inline,the compiler places a copy of the code of that function at each point where the function is called at compile time
- Any change to an inline function could require the function to be recompiled because compiler would need to replace all code once again otherwise it will continue with old functionality.
- Syntax

```
inline return-type function-name(parameters)

{

    // function code

}
```



This pointer

- This pointer is used to hold the address of current object (using which we have called a particular member function).
- This pointer is a const pointer
- This pointer is passed as a hidden argument to non-static member function.
- It is not passed in static member function.
- Friend functions do not have a this pointer, because friends are not members of a class. Only member functions have a this pointer.