

# SPPU-TE-COMP-CONTENT - KSKA Git

## OOP - ASSIGNMENT - 1

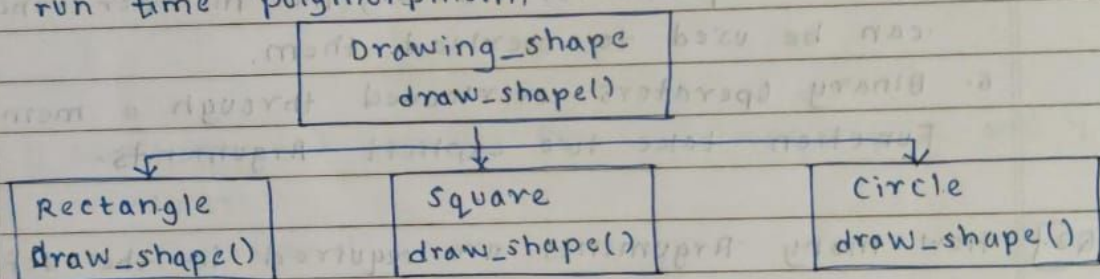
QUESTION:-

Q1] What is Polymorphism?

ANS. Polymorphism is a major feature of Object Oriented Programming Language (OOP) to take more than one form in the programming. It refers to an operation exhibiting different behaviour in different instances. (situations.)

o If we consider the simple case of Addition of two numbers, if we call the function with two numbers then function should perform the Addition. But, if we pass two different strings, then instead of addition, concatenation of string operation should be performed automatically.

o Polymorphism can be compile time polymorphism and run time polymorphism.



o draw\_shape() is to call is totally dependent upon the number of parameters to functions.

o If we are passing radius and centre coordinates than circle type of draw\_shape() should get called.

o If we passing just one side then draw\_shape() for square should get called and so on.

o Using polymorphism we can create a single method and apply it for different objects.

# SPPU-TE-COMP-CONTENT - KSKA Git

Q2.] What are the Rules for Overloading Operator?

ANS. The Rules for Overloading operator are as follows:-

1. Only built-in Operators can be overloaded. New operators can be created i.e. user defined operators cannot be overloaded.
2. Precedence and Associativity of operators cannot be changed.
3. Overloaded operators cannot have default Arguments except the function call operator () which can have default Arguments.
4. Some operators like Assignment operator ("="), Address operator ("&") and comma (",") are by default overloaded.
5. We can't use the Friends Function to overload certain Operators. However, the member Functions can be used to overload them.
6. Binary operators overloaded through a member function take two explicit Arguments.

Q3.] How many Arguments are required in the definition of an overloaded Unary Operator?

ANS. A Unary Operator, eg.: ++a, has one argument a unary means one.

An overloaded unary operator needs no Argument in its method declaration, the objects \* this pointer is the Argument

For Example:-



# SPPU-TE-COMP-CONTENT – KSKA Git

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
using namespace std;

class Coordinates
{
public:
    int x, y; // The field of the class.

    // Constructor to initialize the x and y
    Coordinates (int x, int y)
    {
        this -> x = x;
        this -> y = y;
    }

    // Overloading (-) operator for decrementing x and y
    coordinates.
    void operator -()
    {
        x--;
        y--;
        cout << "\n Decrementing x and y are: " << x
             << " and " << y << endl;
    }
};

// Driver code
int main()
{
```

# SPPU-TE-COMP-CONTENT – KSKA Git

```
Coordinates c(11,13);  
// Using (-) Unary operator for C  
-c;  
return 0;  
}
```