

Q2 b) Every object in C++ has access to its own address through an imp pointer called 'this' pointer. The 'this' pointer is an implicit parameter to all member function.

Example

```
#include <iostream>
using namespace std;
class Variable {
public:
    int x;
    Variable (int x) : x(x) {}
    int maximum (const variable &v) {
        if (this -> x > v.x) {
            return - this -> x;
        }
        return v.x;
    }
};

int main () {
    Variable obj1 (20);
    Variable obj2 (10);
    cout << obj1.maximum (obj2) << endl;
    return 0;
}
```

Name : Ragini Sharma  
Branch : CSE  
ID : B120062

Q3 Function overloading is the process in which different functions with same name are called using different parameter list.

```

class shapes
{
    float A;
    public:
    void area (int a)
    {
        A = 3.14 * a * a;
    }
    void area (int a, int b)
    {
        A = a * b;
    }
}

int main ()
{
    area (5); // calls func1;
    area (2, 3); // calls func2;
    return 0;
}

```

b)

```

#include <iostream>
using namespace std;
class ComplexNumbers
{
    public
    int real;
    int img;
    public:
    void setvalue (int a, int i)
    {
        cin >> real >> img;    real = a;
        img = i;
    }
    void display - complex()

```

Name : Ragini Sharma  
Branch : CSE

ID : B120...

```

    {
        cout << real << " + ( " << " ) i " << endl;
    }
}

void sub (ComplexNumber c1, ComplexNumber c2)
{
    real = c1.real - c2.real;
    imag = c1.imag - c2.imag;
}

}

int main()
{
    ComplexNumber c1, c2, c3;
    cout int r1, i1, r2, i2;
    cout << "Enter initial values: ";
    cin >> r1 >> i1;
    ComplexNumber c, c1, c2, c3;
    c.setvalue (r, i);
    cout << "Enter 1st Complex no. ";
    cin >> r1 >> i1;
    cout << "Enter 2nd Complex no. ";
    cin >> r2 >> i2;
    cout << "Subtraction % ";
    c3 = sub (c1, c2);
    c3.display ();
    return 0;
}

```

Name : Rajini Sharma  
 Branch : CSE  
 ID : B120062