**Assignment – 25 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Classes and Objects**

**Abhishek kumar**

// Q.1.

#include<iostream>

using namespace std;

class Complex

{

    private:

        int a,b;

    public:

    void setData(int x, int y)

    {

        a=x;

        b=y;

    }

    void showData()

    {

        cout << "real = "<< a << " img = "<< b << endl;

    }

};

int main()

{

    Complex c1,c2;

    c1.setData(4,5);

    c1.showData();

    c2.setData(6,8);

    c2.showData();

    return 0;

}

// Q.2.

#include<iostream>

using namespace std;

class Time

{

    private:

       int sec;

       int min;

       int hou;

    public:

    void setTime(int x, int y, int z)

    {

        hou = x;

        min = y;

        sec = z;

    }

    void showTime()

    {

        cout <<hou <<" hr "<<min << " min "<<sec <<" sec "<<endl;

    }

};

int main()

{

    Time t1;

    t1.setTime(3, 45, 20);

    t1.showTime();

}

// Q.3.

#include<iostream>

using namespace std;

class Factorial

{

    private:

      int n;

      int fact;

    public:

    void setN(int x)

    {

        n = x;

    }

    int getN()

    {

        return n;

    }

    int getFactorial()

    {

        return fact;

    }

    void calculateFact(int n)

    {

        int f=1,i;

        if(n<=0)

        {

            fact = 1;

            return;

        }

    }

    void calculateFactorial()

    {

        int i,f=1;

        for(i=1; i<=n; i++)

        {

            f = f\*i;

        }

           fact = f;

     }

};

int main()

{

    Factorial f,f2;

    f.setN(5);

    f2.setN(6);

    f.calculateFactorial();

    f2.calculateFactorial();

    cout << "Factorial 5 is = "<< f.getFactorial() <<endl << "Factorial 6 is = "<< f2.getFactorial()<< endl;

}

// Q.4.

#include <iostream>

using namespace std;

class LargestNumber

{

    private:

       int a,b,c;

    public:

    void input()

    {

        cout << "Enter the numbers:";

        cin >> a >> b >> c;

    }

    void cal()

    {

        if(a>b)

        {

            if(a>c)

               cout << "Greatest "<<a;

            else

               cout << "Greatest "<<c;

        }

        else

        {

            if(b>c)

                cout << "Greatest "<<b;

            else

                cout << "Greatest "<<c;

        }

    }

};

int main()

{

    LargestNumber l;

    l.input();

    l.cal();

}

// Q.5.

#include<iostream>

using namespace std;

class ReverseNumber

{

    private:

        int n,r=0;

    public:

    void setinput()

    {

        cout << "Enter the numbers:";

        cin >> n;

    }

    void cal()

    {

        while(n!=0)

        {

            r = (r\*10)+(n%10);

            n/=10;

        }

        cout << "Reverse number is " << r;

    }

};

int main()

{

    ReverseNumber s;

    s.setinput();

    s.cal();

}

// Q.6.

#include<iostream>

using namespace std;

class Square

{

    private:

       int side;

    public:

    void setSquare(int l)

    {

        side = l;

    }

    int Length\_breath()

    {

        return side\*side;

    }

};

int main()

{

    Square s;

    s.setSquare(5);

    cout << "Area of Square " << s.Length\_breath() << endl;

}

// Q.7.

#include<iostream>

using namespace std;

class Greatest

{

    private:

       int a,b,c;

    public:

    void input()

    {

        cout << "Enter the numbers:";

        cin >> a >> b >> c;

    }

    void calculate()

    {

        if(a>b)

        {

            if(a>c)

              cout << "Greatest is "<< a;

            else

              cout << "Greatest is "<< c;

        }

        else

        {

            if(b>c)

               cout << "Greatest is "<< b;

            else

               cout << "Greatest is "<< c;

        }

    }

};

int main()

{

    Greatest g;

    g.input();

    g.calculate();

}

// Q.8.

#include<iostream>

using namespace std;

class Rectangle

{

    private:

       float lenght , breath;

    public:

    void setLengthBreath(float l, float b)

    {

        lenght = l;

        breath = b;

    }

    float rectangle()

    {

        return (lenght\*breath);

    }

};

int main()

{

    Rectangle r;

    r.setLengthBreath(3,4);

    cout << "Area of Rectangle:"<< r.rectangle() << endl;

}

// Q.9.

#include<iostream>

using namespace std;

class Circle

{

    private:

       float radius;

    public:

    void setCircle(float r)

    {

        radius = r;

    }

    float circle()

    {

        return 3.14\*radius\*radius;

    }

};

int main()

{

    Circle c;

    c.setCircle(4);

    cout << "Area of Circle " << c.circle() << endl;

}

// Q.10.

#include<iostream>

using namespace std;

class Area

{

        float side,length,breath,radius;

    public:

    void setSide(float s)

    {

        side = s;

    }

    void setLengtBreath(float l, float b)

    {

        length = l;

        breath = b;

    }

    void setRadius(float r)

    {

        radius = r;

    }

    float sqare()

    {

        return side\*side;

    }

    float rectangle()

    {

        return length\*breath;

    }

    float circle()

    {

        return 3.14\*radius\*radius;

    }

};

int main()

{

    Area a;

   a.setSide(5);

   a.setLengtBreath(3, 4);

   a.setRadius(4);

   cout <<"Area of Square is "<<a.sqare() <<endl;

   cout <<"Area of Rectangle is "<<a.rectangle() <<endl;

   cout <<"Area of Circle is "<<a.circle() <<endl;

}

\*/