```cpp
//////////////////////////////////////////////////////////////////////////////
// Faculty of Computing, Universiti Teknologi Malaysia
// SCSJ1023- Programming Technique II
// Semester 1, 2017/2018
// Final Exam, Paper 2, Question 2  (Problem Solving)
// Prepared by: Jumail Bin Taliba (jumail@utm.my)
// 5 November 2017
//      The program itself = 60 marks
//      Able to run and proper writing 5 marks
//      Total = 65 marks
//////////////////////////////////////////////////////////////////////////////

#include<iostream>
using namespace std;

#define PI 3.1415

// 6m (total). 1m each
class Point{
    private:
        double x, y;
    public:
        Point(double _x=0, double _y=0){set(_x,y);}
        void set(double _x, double _y){x =_x; y=_y;}
        double getX()const {return x;}
        double getY()const {return y;}
};

// 5m (total). 1m each.
class Shape{
    public:
        Shape(){}
        virtual double getArea() const=0;  // virtual will contribute to criteria 'Polymor
        virtual void read()=0;
        virtual void print()=0;
};


//  15 marks (total)
class Circle:public Shape{  // 1m + 1m    // 1m goes to 'inheritance'
    private:
        Point center;  // 1m  goes to composition
        double radius; // 1m

    public:
        Circle():center(0,0){radius=0;}  // 1m

        double getArea() const{return PI*radius*radius;}  // 2m

        void read(){  // 1m goes to this class definition , the following marks go to 'pol
            double cx, cy;
            cout << "Enter the center coordinates of the circle (x y)=> ";
            cin >> cx >> cy;  // 1m
            center.set(cx,cy); // 1m
            cout << "Enter the circle's radius=> ";
            cin >> radius;  // 1m
        }

        void print(){  // 1m goes to this class definition , the following marks go to 'po
            cout << "Circle's center: X=" << center.getX() << "  Y=" << center.getY() << e
            cout << "Circle's radius =" << radius << endl;  // 1m
            cout << "Circle's area   =" << getArea() << endl; // 1m
        }
};

// 19 marks (total)
class Rectangle:public Shape{  // 1m + 1m   // 1m goes to inheritance
    private:
        Point topLeft, bottomRight;  // 1m goes to composition
        double getWidth() const { return bottomRight.getX()-topLeft.getX();}  // 2m
        double getHeight() const { return topLeft.getY()-bottomRight.getY();} // 2m

    public:
        Rectangle():topLeft(0,0), bottomRight(0,0) {} // 1m

        double getArea() const{return getWidth() * getHeight();}  // 2m

        void read(){  // 1m  // 1m goes to this class definition , the following marks go
            double x1, y1, x2, y2;

            cout << "Enter the top left corner coordinates of the rectangle (x y) => ";
            cin >> x1 >> y1;  // 0.5m
            cout << "Enter the bottom right corner coordinates of the rectangle (x y) => "
            cin >> x2 >> y2; // 0.5m

            topLeft.set(x1,y1); // 0.5m
            bottomRight.set(x2,y2);  // 0.5m
        }

        void print(){ // 1m // 1m goes to this class definition , the following marks go t
            cout << "Rectangle's top left corner: X=" << topLeft.getX() << "  Y=" << topLe
            cout << "Rectangle's bottom right corner: X=" << bottomRight.getX() << "  Y="
            cout << "Rectangle's width  = " << getWidth()<< endl;  // 1m
            cout << "Rectangle's height = " << getHeight()<< endl; // 1m
            cout << "Rectangle's area   = " << getArea()<< endl;  // 1m
        }
};

int menu()
{
    int choice;

    cout << endl << endl;
    cout << "==========[MENU]============" << endl <<endl;
    cout << "1. Add a shape" << endl;
    cout << "2. Print all shapes" << endl;
    cout << "3. Calculate total area" << endl;
    cout << "4. Exit" << endl << endl;

    cout << "Enter your choice => ";
    cin >> choice;
    cout << endl;

    return choice;
}

int main()
{
    int shapeCount=0;   // 0.5m  goes to 'using single array'
    Shape *shapes[20];  // 1m  goes to 'using single array'
```

```cpp

    int command = menu();  // 1m goes to 'using menu'
    int shapeType;
    Shape *newShape;

    while (command!=4){ // 1m goes to 'using menu'

        switch (command){
            case 1:
                cout << "What type of shape you want to enter? " << endl
                     << "     1. Circle" <<endl
                     << "     2. Rectangle" <<endl<< endl;

                cout << "Your choice => ";
                cin >> shapeType;   // 1m  goes to using menu
                cout << endl;

                if (shapeType==1) newShape = new Circle; else newShape=new Rectangle;  //
                newShape->read();  // 1m goes to polymorphism
                shapes[shapeCount]=newShape; // 1m  goes to 'using array
                shapeCount++;  // 0.5m  goes to 'using array'

                break;

            case 2:
                for (int i=0; i<shapeCount; i++){  // 1m   goes to printing all shapes
                    cout << "Shape #" << (i+1) << endl;
                    shapes[i]->print();   // 1m
                    cout << endl << endl;
                }

                break;

            case 3:
                double totalArea = 0;  // 1m  goes to calculating total area
                for (int i=0; i<shapeCount; i++) // 1m
                    totalArea += shapes[i]->getArea(); // 1m
                cout << "Total Area= " << totalArea << endl; // 1m

                break;
        }

        command = menu();
    }

    return 0;
}
```