EXPERIMENT NO. 9

Problem Statement 1: To create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle subclass that each one of the classes extends the Class Shape. Each one of the classes contains only the method printArea() that prints the area of Shape.

Program:

abstract_class.php

```
<?php

// Abstract class named Shape
abstract class Shape
{
  protected $length;
  protected $width;

  public function __construct($length, $width)
  {
    $this->length = $length;
    $this->width = $width;
  }

  abstract public function printArea();
}

// Rectangle subclass
class Rectangle extends Shape
{
  public function printArea()
  {
    $area = $this->length * $this->width;
    echo "The area of the rectangle is: " . $area . "<br>";
  }
}

// Triangle subclass
class Triangle extends Shape
{
  public function printArea()
  {
    $area = ($this->length * $this->width) / 2;
    echo "The area of the triangle is: " . $area . "<br>";
  }
}
```

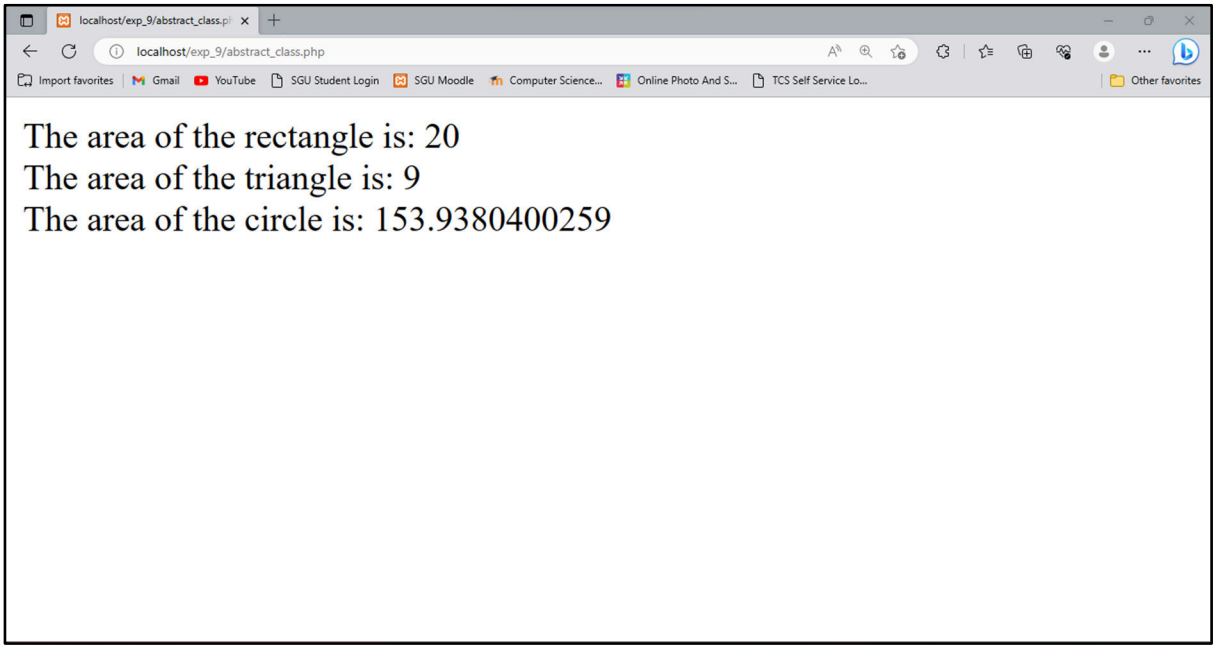
```
// Circle subclass
class Circle extends Shape
{
  public function printArea()
  {
    $area = pi() * ($this->length ** 2);
    echo "The area of the circle is: " . $area . "<br>";
  }
}

// Instantiate and use the classes
$rectangle = new Rectangle(4, 5);
$rectangle->printArea();

$triangle = new Triangle(3, 6);
$triangle->printArea();

$circle = new Circle(7,null);
$circle->printArea();
?>
```

Output:



Problem Statement 2: Write a program to create an interface named Shape that contains empty method named Area () and perimeter(). Provide two classes named Triangle and Circle such that each one of the classes implement Shape. Each one of the classes contains only the method Area () and perimeter().that prints the area and perimeter() of the given shape.

Program:

interface.php

```
<?php

// Define the Shape interface
interface Shape
{
  public function area();
  public function perimeter();
}

// Define the Triangle class that implements Shape
class Triangle implements Shape
{
  private $a, $b, $c;

  public function __construct($a, $b, $c)
  {
    $this->a = $a;
    $this->b = $b;
    $this->c = $c;
  }

  public function area()
  {
    $s = ($this->a + $this->b + $this->c) / 2;
    return sqrt($s * ($s - $this->a) * ($s - $this->b) * ($s - $this->c));
  }

  public function perimeter()
  {
    return $this->a + $this->b + $this->c;
  }
}

// Define the Circle class that implements Shape
class Circle implements Shape
{
  private $radius;

  public function __construct($radius)
```

```
  {
    $this->radius = $radius;
  }

  public function area()
  {
    return pi() * pow($this->radius, 2);
  }

  public function perimeter()
  {
    return 2 * pi() * $this->radius;
  }
}

// Create a Triangle object and print its area and perimeter
$triangle = new Triangle(3, 4, 5);
echo "Triangle area: " . $triangle->area() . "<br>";
echo "Triangle perimeter: " . $triangle->perimeter() . "<br>";

// Create a Circle object and print its area and perimeter
$circle = new Circle(5);
echo "Circle area: " . $circle->area() . "<br>";
echo "Circle perimeter: " . $circle->perimeter() . "<br>";
?>
```

Output: