**Determine the value of PI:**  
Design and Implement necessary classes so that the value of PI is calculated by the following code.

Constraints:

* All the attribute of class Point and Circle should be private.
* getRandomPoint() returns a random point in rectangle [(0, 0), (1, 1)].
* Follow the class instruction for random number generation.

|  |
| --- |
| int main() {  srand(time(0));  Circle C( Point( .5, .5 ), .5 );  int nt = 1000000, nc = 0;  for(int i=0; i<nt; i++) {  if(C.isIn( getRandomPoint() ))nc++;  }  printf("Value of PI is %lf\n", 4.0\*nc/nt);  return 0;  } |

\*This method is called Monte Carlo Algorithm for determining area/volume of complex object.

|  |
| --- |
| #include<iostream>  #include<time.h>  #include<stdlib.h>  using namespace std;  double getRand(){  double value=((double)rand()/(double)RAND\_MAX);  return value;  }  int main(){  srand(time(0));  for(int i=0;i<100;i++){  cout<<getRand()<<endl;  }  return 0;  } |

|  |
| --- |
| #include<iostream>  #include<time.h>  #include<stdlib.h>  #include<stdio.h>  using namespace std;  class Point{  public:  double x, y;  Point(){}  Point(double \_x, double \_y){x=\_x; y=\_y;}  };  class Circle{  public:  Point c;  double r;  Circle(){}  Circle(Point \_c, double \_r) {c=\_c; r=\_r;}  bool isIn(Point p) {  return (c.x-p.x)\*(c.x-p.x) + (c.y-p.y)\*(c.y-p.y) <=r\*r;  }  };  double getRand(){  double value=((double)rand()/(double)RAND\_MAX);  return value;  }  Point getRandomPoint() {  return Point( getRand(), getRand() );  }  int main() {  srand(time(0));  Circle C( Point( .5, .5 ), .5 );  int nt = 10000000, nc = 0;  for(int i=0; i<nt; i++) {  if(C.isIn( getRandomPoint() ))nc++;  }  printf("Value of PI is %lf\n", 4.0\*nc/nt);  return 0;  } |