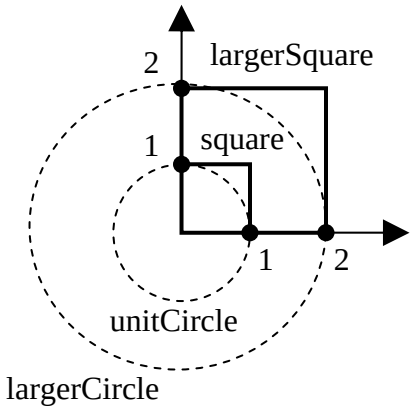
OBJECT ORIENTED PROGRAMMING

1st Midterm Exam

(There are 2 Questions. 90-Minutes Exam, Give your answer inside the corresponding box)

Student ID	Name	Signature

Q.1) (60) You will design two classes in order to model two geometrical shapes, Circle and Rectangle, using object oriented programming approach. A typical application code is given below.
Write only required methods.




```
Circle unitCircle;
Circle largerCircle(0.0,0.0,2.0);
Rectangle square;
Rectangle largerSquare(0.0,2.0,2.0,0.0);
double f= unitCircle.area();
double g= unitCircle.circumference();
double h= square.area();
double j= square.circumference();
if (unitCircle>largerCircle) {…}
if (unitCircle==largerCircle) {…}
if (square>largerSquare) {…}
if (square==largerSquare) {…}
if (unitCircle>square) {…}
if (square==largerCircle) {…}
double left,right,top,bottom;
square.getPosition(left,top,right,bottom);
double x,y,r;
unitCircle.getPosition(x,y,r);
largerSquare.setPosition(left,top,right,bottom);
largerCircle.setPosition(x,y,r);
```

Comparisons between rectangles and circles are computed through areas.

You may use the back of the page, but not elsewhere

Q.2) (40) What is the output of the following C++ code?

```
class Storage {
    int *hold;
  public:
    Storage (int h=0) {
      hold= new int(h);
      cout << endl << "constructor (Storage): Holding " << h ;
    }
    ~Storage () {
      cout << endl << "destructor (Storage): Releasing " << *hold ;
      delete hold;
    }
    Storage (const Storage &source) {
      hold= new int(*source.hold);
      cout << endl << "copy constructor (Storage): Holding " << *hold ;
    }
    Storage& operator= (const Storage& right) {
      cout << endl << "operator= (Storage): Releasing " << *hold ;
      delete hold;
      hold= new int(*right.hold);
      cout << endl << "operator= (Storage): Holding " << *hold ;
    }
    int& operator()() const {
      cout << endl << "operator() (Storage)" ;
      return *hold;
    }
} ;
class Array {
    Storage *array ;
    int size;
  public:
    Array(int *a,int length){
      cout << endl << "constructor (Array)" ;
      array= new Storage[length];
      for (int i=0;i<length;i++){
         array[i]()= a[i];
      }
      size= length;
    }
    int& operator[] (int i) const {
      cout << endl << "operator[] (Array)" ;
      return array[i]();
    }
    ~Array(){
      cout << endl << "destructor (Array) ";
      delete[] array;
     }
} ;
```

```
   int main (int argc, char* argv[]){
①     int myArray[]={549,3615,4629};
②     int length=sizeof(myArray)/sizeof(int);
③     Array myObject(myArray,length);
④     for (int i=0;i<length;i++)
⑤         cout << endl << myObject[i] ;
⑥     return 0;
⑦  }
```

Line Number	Output