

Name:

Student ID:

Signature:

Langara College

CPSC 1181 – Midterm #1 (Weeks 1-4) – Section 002

June 6, 2017, 8:30am

Hash: xxxxxxxxxx

Instructor: Jeremy Hilliker

Duration 1h45m (no extensions)

- Permitted aids: 1 single-sided letter/a4 reference sheet, hand written by the student taking the exam; no other aids
 - Reference sheet must be submitted with your exam (failure to do so will result in a penalty)
 - If you do not have a reference sheet, you must alert the invigilator before the first half of the exam expires.
 - You may remove the Appendices from the exam, but they too must be handed in
- Turn off all electronic devices
- Store all personal belongings out of reach
 - Ask the invigilator for permission before retrieving any personal belongings
- Strictly no speaking to other students during the exam
- The work submitted for this exam must be entirely your own
- Answer all questions in the provided spaces
 - Use backs of facing pages if more space is needed
- The exam is over when announced by the invigilator
- Print your name and student ID on the cover sheet of this exam
- Follow best programming practices to receive full marks
 - You may omit comments (except where noted)

Q	Max	Awarded
1	10	
2	12	
3	9	
4 (pg 6)	12	
4 (pg 7)	15	
4 (pg 8)	7	
5	6	
Total	71	

Appendix 4

```
public class Robot {  
    private final static int WARRANTY = 100000;  
    private static int nextID = 101;  
  
    private final int id;  
    private int odometer;  
  
    /** Creates a Robot with a unique ID and odometer set to 0.  
    */  
    public Robot() {  
        id = nextID++;  
    }  
  
    /** Moves the Robot. Odometer' = odometer + howFar  
    @param howFar the distance to move the robot.  
    */  
    public void move(int howFar) {  
        odometer += howFar;  
    }  
  
    /** Gets the robot's unique ID.  
    @return the robot's unique ID.  
    */  
    public int getID() {  
        return id;  
    }  
  
    /** Get the robot's current odometer.  
    @return how far the robot has moved over its life.  
    */  
    public int getOdometer() {  
        return odometer;  
    }  
  
    /** Determines if the robot is still under warranty.  
    @return true if the robot is still under warranty, false otherwise  
    */  
    public boolean isUnderWarranty() {  
        return odometer < WARRANTY;  
    }  
  
    /** Determines if this robot is the same robot as another.  
    @param o the other robot to compare to  
    @return true if the other robot is the same as this one,  
            False otherwise.  
    */  
    public boolean equals(Robot o) {  
        return this.id == o.id;  
    }  
}
```

Appendix API

Graphics2D
Graphics create()
void dispose()
void draw(Shape s)
void drawLine(int x1, int y1, int x2, int y2)
void drawRect(int x, int y, int width, int height)
void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)
void drawPolygon(Polygon p)
void drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
AffineTransform getTransform()
void rotate(double theta)
void rotate(double theta, double x, double y)
void scale(double sx, double sy)
void setColor(Color c)
void setStroke(Stroke s)
void setTransform(AffineTransform Tx)
void translate(int x, int y)
Rectangle
Rectangle(int x, int y, int width, int height)
void translate(int dx, int dy)
Polygon
Polygon()
Polygon(int[] xpoints, int[] ypoints, int npoints)
void addPoint(int x, int y)
void translate(int deltaX, int deltaY)
Collections
HashSet<E>()
ArrayList<E>()
void add(E e)
void addAll(Collection<? extends E> c)
void clear()
boolean contains(Object o)
boolean containsAll(Collection<?> c)
boolean equals(Object o)
boolean isEmpty()
Iterator<E> iterator()
boolean remove(Object o)
boolean removeAll(Collection<?> c)
boolean retainAll(Collection<?> c)
int size()
Iterator<E>
boolean hasNext()
E next()
void remove()