The University of Western Ontario
Department of Computer Science

Computer Science 026a Midterm Exam
October 27, 2007
2 hours

Print your name: ____

Student number: _______________________________________

Instructions:

- **Fill in your name and student number above immediately.**
- **On the Scantron form, fill in your name and signature in the spaces provided at the top. Encode your student number in the area indicated by STUDENT NUMBER.**
- You have 2 hours to complete the exam. **There will be NO extra time given to fill in the Scantron form at the end of the exam.**
- The exam is out of a possible 110 marks (Part 1 is out of 60, Part 2 is out of 50).
- **Part 1** of the exam consists of Multiple Choice questions. Circle your answers on this exam paper and fill in your answers on the provided Scantron form.
- **Part 2** consists of questions for which you will provide written answers: Write your answers in the spaces provided in this exam paper.
- The marks for each individual question are given. Allow approximately 1 minute per mark on average.
- There is a page for rough work at the very back of the exam. You may detach it if you wish, but hand it in with the rest of the exam paper.

Documentation concerning the methods you may need for the classes World, Turtle, Picture, Pixel, FileChooser and Color is provided at the back of the exam.

DO NOT TURN THIS PAGE UNTIL DIRECTED TO DO SO.

Part 2 Marks

1	
2	
3	
4	
5	
6	
7	
Total	

Part 1: Multiple Choice

Choose the best answer from the choices given. Choose only ONE answer. Each question is worth 1 mark unless otherwise indicated.

You may assume that there are no syntax errors intended in any of the code provided in the questions of Part 1.

1. A compiler will
   (a) find syntax errors
   (b) fix syntax errors
   (c) find logic errors
   (d) more than one of the above
   (e) none of the above answers

2. An algorithm should be written
   (a) in Java
   (b) in machine language
   (c) without syntax errors
   (d) in a natural language such as English
   (e) none of the above answers

3. In order to run a Java program,
   (a) it must be compiled
   (b) it must have a main method
   (c) it must have at least one other method
   (d) exactly two of (a), (b), (c)
   (e) all of (a), (b), (c)

4. What will be displayed on the screen by the following code segment?

```
String word = “Hello”;
System.out.println(word);
```

   (a) word
   (b) Hello
   (c) “Hello”
   (d) “Hello”;
   (e) none of the above

5. In the previous question, `println` is the name of a
   (a) variable
   (b) method
   (c) class
   (d) object
   (e) none of the above

6. What will be output by the following Java statement?

```
System.out.println((double)(12/5));
```

(a) 2.4
(b) 2.0
(c) 2
(d) 2.5
(e) none of the above

7. What will be output by the following Java statement?

```
System.out.println((double)12/5);
```

(a) 2.4
(b) 2.0
(c) 2
(d) 2.5
(e) none of the above

8. What will be output by the following Java statement?

```
System.out.println((int)(12.0/5));
```

(a) 2.4
(b) 2.0
(c) 2
(d) 2.5
(e) none of the above

9. What will be output by the following Java statement?

```
System.out.println((int)12.0/5));
```

(a) 2.4
(b) 2.0
(c) 2
(d) 2.5
(e) none of the above

10. (2 marks) Suppose we have defined

```
int m = 18, n = 4;
```

The value of the expression `m / n + m % n` is

(a) 6.5
(b) 6
(c) 6.0
(d) 100
(e) none of the above

11. (2 marks) Suppose we have defined

```
int n = 4;
double x = 2.5;
```

The value of the expression `5 * x – n / 5` is

(a) 12.5
(b) 11.7
(c) 1.7
(d) 11.5
(e) none of the above

12. Suppose we have defined

```
int num1 = 50, num2 = 50;
```

The value of the expression `(num1 == num2)` is

(a) true
(b) false
(c) 50
(d) 0
(e) none of the above

13. What will be printed by the following statement?

```
System.out.println(“x\\y”);
```

(a) x\\y
(b) x\y
(c) xy
(d) this statement will not compile
(e) none of the above answers

14. (2 marks) What will be the value of x after the following code segment is executed?

```
int x = 3;
x = x + x;
x = x + x;
```

(a) 3
(b) 6
(c) 9
(d) 12
(e) none of the above

15. (2 marks) What will be the value of y after the following code segment is executed?

```
int x = 3,  y = 5;
x = y;
y = x;
```

(a) 3
(b) 5
(c) 6
(d) 10
(e) none of the above

Consider the following Java program for the next 9 questions (you may assume it compiles):

```
 public class Midterm {
     public static void main (String [] args) {
       final int SIZE = 10;
       int num = 75;
       World w1 = new World();
       Turtle t1 = new Turtle(w1);
       t1.forward(num);
       t1.drawSquare(SIZE);
     }
  }
```

16. The identifier `Midterm` in the above code is the name of
    (a) a method
    (b) an object
    (c) a class
    (d) a file
    (e) none of the above

17. The Java keyword `void` in the above code means that
    (a) the method `main` has no parameters
    (b) the method `main` returns the integer 0
    (c) the method `main` returns the keyword `null`
    (d) the method `main` returns nothing
    (e) none of the above

18. Which of the following statements is/are true about the method `main` in the above code?
    (a) It is a class method.
    (b) It is an object method.
    (c) It is executed by using the statement `Midterm.main();`
    (d) Exactly two of (a), (b), (c)
    (e) None of the above answers

19. The identifier `num` in the above code is a
    (a) reference variable
    (b) variable of primitive type
    (c) constant
    (d) name of a method
    (e) none of the above

20. The identifier `SIZE` in the above code is a
    (a) reference variable
    (b) variable of primitive type
    (c) constant
    (d) name of a method
    (e) none of the above

21. The identifier `w1` in the above code is a

    (a) reference variable
    (b) variable of primitive type
    (c) constant
    (d) name of a method
    (e) none of the above

22. Which of the following statements is/are true about the keyword **`new`** in the above code?

    (a) It refers to the current object.
    (b) It is the name of a method.
    (c) It is the name of a variable.
    (d) It creates a new object.
    (e) More than one of the above statements are true.

23. Which of the following in the above code is/are parameters to methods?

    (a) num
    (b) w1
    (c) t1
    (d) exactly two of (a), (b), and (c)
    (e) all of (a), (b), and (c)

24. Which of the following in the above code is/are object methods?

    (a) World
    (b) Turtle
    (c) forward
    (d) exactly two of (a), (b), and (c)
    (e) all of (a), (b), and (c)

25. Suppose we have an array declared by

```
int [] intArr = {2,4,6,8,10};
```

What value is stored in `intArr[1]` ?

    (a) 1
    (b) 2
    (c) 4
    (d) 0
    (e) none of the above

26. How can we refer to the last element in the array of the previous question?

    (a) intArr[10]
    (b) intArr[intArr.length]
    (c) intArr[intArr.length – 1]
    (d) intArr[0]
    (e) none of the above

27. (2 marks) Suppose we want to store the value 100 in each of the elements of the array `intArr2` declared by:

```
 int [] intArr2 = new int[12];
```

Which of the following `for` loops would do this correctly?

(a) for (int k = 1; k <= 12; k++)
    intArr2[k] = 100;

(b) for (int k = 0; k <= 12; k++)
    intArr2[k] = 100;

(c) for (int k = 1; k <= 11; k++)
    intArr2[k] = 100;

(d) for (int k = 0; k <= 11; k++)
    intArr2[k] = 100;

(e) none of the above

28. In the previous question, `intArr2` is a
(a) reference variable
(b) variable of primitive type
(c) constant
(d) name of a method
(e) none of the above

Consider the following code segment. Use it to answer the next 4 questions.

```
int n1, n2 = 0;
double x = 12.34;
String test;
World w1 = new World();
Turtle t1 = new Turtle(w1);
Turtle t2 = t1;
```

29. How many variables are declared in this code segment?
(a) 0
(b) 5
(c) 6
(d) 7
(e) none of the above

30. How many objects are created in this code segment?
(a) 0
(b) 1
(c) 2
(d) 3
(e) none of the above

31. The variables `t1` and `t2` in the above code segment refer to different objects of the Turtle class.

    (a) True
    (b) False

32. Which if the following is/are true about the variable `test` in the above code?

    (a) It is of a primitive data type.
    (b) It is initialized to zero (0).
    (c) It is initialized to null.
    (d) More than one of the above
    (e) None of the above answers

Use the following Java code segment for the next 4 questions.

```
Picture picture = new Picture(getMediaPath(“caterpillar.jpg”));
Pixel [] pixelArray = picture.getPixels();
int value = 0;
int count = 0;
while (count < pixelArray.length)
{
   pixelArray[count].setRed(value);
   pixelArray[count].setBlue(value);
   pixelArray[count].setGreen(value);
   count = count + 1;
}
picture.show();
```

33. (2 marks) Which of the following statements most correctly describes what the above code does to the Picture object referenced by `picture`?

    (a) It makes all the pixels be white.
    (b) It makes all the pixels be gray.
    (c) It makes all the pixels be black.
    (d) It does not change the color of any of the pixels.
    (e) None of the above

34. The variable `pixelArray` in the above code refers to

    (a) a Picture object
    (b) a Pixel object
    (c) a one-dimensional array
    (d) a two-dimensional array
    (e) none of the above

35. The number of pixels in the Picture object referenced by `picture` in the above code is the value

    (a) pixelArray.length()
    (b) pixelArray.length
    (c) pixelArray[length]
    (d) pixelArray[length – 1]
    (e) none of the above

36. The above code segment changes the image stored in the file *caterpillar.jpg*.

(a) True
(b) False

37. (3 marks) What will be output by the following code segment?

```
int sum = 0;
int count = 1;
while (count < 5)
{   sum = sum + count;
   count = count + 2;
}
System.out.println(sum);
```

(a) 10
(b) 15
(c) 4
(d) 9
(e) none of the above

38. (3 marks) What will be output by the following code segment?

```
int sum = 0;
int count = 5;
while (count >1)
{
    sum = sum + count;
    count = count + 2;
}
System.out.println(sum);
```

(a) 10
(b) 15
(c) 4
(d) 9
(e) none of the above

39. (2 marks) What will be output by the following code segment?

```
Picture pic1 = new Picture(100,100);
Picture pic2 = new Picture(200,200);
pic2 = pic1;
System.out.println(pic2.getHeight());
```

(a) 100
(b) 200
(c) 300
(d) 400
(e) none of the above

40. (3 marks) Suppose we have the following method defined in the Turtle class:

```
 public void someMove(int length)
{
    this.forward(length);
    this.turnRight();
}
```

Suppose we have the following sequence of statements executed in the Interactions pane:

```
 World w1 = new World();
Turtle t1 = new Turtle(100,100,w1);
t1.someMove(25);
System.out.println(t1.getXPos() + “ “ + t1.getYPos());
```

What will be output when the above code sequence is executed?

(a) 100 25
(b) 100 75
(c) 25 100
(d) 75 100
(e) none of the above

41. (2 marks) Consider the following code segment:

```
Picture picture = new Picture(getMediaPath(“caterpillar.jpg”));
int x = 0;
int y = picture.getHeight() – 1;
Pixel pixel = picture.getPixel(x,y);
```

The location of the pixel referenced by the variable `pixel` is:

(a) at the top left of the picture
(b) at the top right of the picture
(c) at the bottom left of the picture
(d) at the bottom right of the picture
(e) none of the above

42. The statement that would change the color of the pixel of the previous question to black is

(a) Pixel.setColor(java.awt.Color.BLACK);
(b) pixel.setColor(java.awt.Color.BLACK);
(c) pixel.color = java.awt.Color.BLACK;
(d) picture.setColor(x,y, java.awt.Color.BLACK);
(e) none of the above

43. Consider the following code segment, assuming that the method with the header
`public void copyPictureTo(Picture sourcePicture, int xstart, int ystart)`
has been added to the Picture class:

```
Picture pic1 = new Picture(900,900);
Picture pic2 = new Picture(FileChooser.pickAFile());
pic1.copyPictureTo(pic2,100,100);
```

When this code segment is executed, the picture referenced by `pic1` will be copied onto the picture referenced by `pic2`, with the top left corner of the picture of `pic1` at (100,100) of the picture of `pic2`.

(a) True
(b) False

44. (3 marks) What will be output by the following code segment?

```
for (int row = 1; row <= 3; row ++)
{
    for (int count = 1; count <= (4 – row); count ++)
    {
      System.out.print(“*”);
     }
   System.out.println();
}
```

(a)
```
*
**
***
```

(b)
```
***
**
*
```

(c)
```
***
***
***
```

(d)
```
*
*
*
```

(e) none of the above

Part 2: Write your answers in the spaces provided below each question.

1. (7 marks) Consider the following Java code segment:

```
String item = new String("Book");
double price = 100.0;
double taxRate = 0.10;
System.out.println(taxRate);
double finalPrice = price + price*taxRate;
System.out.println("Book price before tax: " + price);
System.out.println("Book price after tax: " + finalPrice);
 System.out.print("Tax paid: ");
System.out.println(price*taxRate);
```

   What does this code segment print to the screen?

2. (5 marks) The following Java code segment makes use of a `while` loop to compute the sum of all numbers between 1 and 100 that are divisible by 3. In the space provided, you should change it to do the same thing but use a `for` loop.

```
int sum = 0;
int i = 3;
while (i < 100)
{
   sum = sum + i;
   i = i + 3;
 }
```

3. (10 marks) Recall the Turtle World examples from class, labs and Assignment 1. Provide a sequence of Java statements that will create a new world and a new turtle within that world, and will then instruct the turtle to create a block character of the letter U, for which the left hand and right hand sides are each 50 units high and the bottom line is 30 units wide. You only need to provide the statements required to do this in the Interactions pane of DrJava; you do not need to contain these statements within methods or a program. The turtle should start and end at the same point, facing in the same direction as it started.

4. (5 marks) Suppose that the method drawSquare() has been implemented in the Turtle class as follows:

```
public void drawSquare(int sideLength)
{
  this.penDown();
  for (int k = 1; k <= 4; k++ )
  {
    this.turnLeft();
    this.forward(sideLength);
  }
}
```

Provide a sequence of Java statements that will create a new world and a new turtle within that world, and will then instruct the turtle to draw a square with sides of length 50, move forward by 20 units, and draw another square with sides of length 20.

5. (8 marks) The following Java method added to the Picture class is intended to change the green in the picture by the amount passed in as the parameter. It has several logic errors. (The code compiles correctly, but does not behave as expected when it is run.) Correct each error; there is at most one error in a line. In the space below the method, write the *line number* for each line with an error, and then *rewrite the line with the correction made*. (Line numbers are to the left of each line of code.)

```
1     public void changeGreen(double amount)
2     {
3          Pixel[] pixelArray = this.getPixels();
4          Pixel pixel = null;
5          int value = 0;
6
7          //loop through all the pixels
8          for (int i = 0; i <= pixelArray.length; i++)
9          {
10              // get the current pixel
11              pixel = pixelArray[value];
12
13              //get its green value
14              value = pixel.getGreen();
15
16              //change the green value
17              value = (int) (value * amount);
18
19              //set the new green value
20              pixel.setGreen(i);
21
22              i = i + 1;
23         }
24    }
```

<u>Error in Line number</u> **<u>Correction</u>**

6. (4 marks) The following Java method added to the Picture class changes a picture to grayscale. This is done in *row order* i.e., the rows are done one by one. How would you change the code to use *column order*, i.e. so that the columns are done one by one? You do not need to rewrite the whole method; in the space provided below, write the line number for each line that needs to be changed, and then rewrite the line with the changes made. (Line numbers are to the left of each line of code.)

```
1     public void grayscale()
2     {
3        int intensity = 0;
4        Color color = null;
5        Pixel pixel = null;
6
7        for (int y = 0; y < this.getHeight(); y++)
8        {
9           for (int x = 0; x < this.getWidth(); x++)
10          {
11           //get the current pixel
12           pixelObj = this.getPixel(x,y);
13
14           //get the current color
15           Color = pixel.getColor();
16           //convert to grayscale
17           intensity = (int) ((pixel.getRed()+
18                        pixel.getGreen()+ pixel.getBlue())/3);
19           Pixel.setColor(new Color(intensity,intensity,intensity));
20
21          }
22       }
23    }
```

<u>**Line number**</u>   <u>**Changed Line**</u>

7. (11 marks) The following Java method added to the Picture class changes a picture to its negation. Suppose that we want to modify the method so that it takes four integer parameters `startX`, `startY`, `endX`, and `endY`, where `startX`, `startY` are the coordinates of the position that the negation should start with, and `endX`, `endY` are the coordinates of the position where negation ends. This means that only a part of the picture will be negated by this method. You should modify the provided method to support this. There are **exactly three** lines that need to be changed. In the space provided below, write the line number for each line that needs to be changed, and then rewrite the line with the changes made. (Line numbers are to the left of each line of code.)

```
1  public void negate()
2    {
3    Color color = null;
4    Pixel pixelObj = null;
5
6    //loop through the rows
7    for (int y = 0; y < this.getHeight(); y++)
8    {
9          // loop through the columns
10         for (int x = 0; x < this.getWidth(); x++)
11         {
12           pixelObj = this.getPixel(x,y);
13
14           int redValue = pixelObj.getRed();
15           int greenValue = pixelObj.getGreen();
16           int blueValue = pixelObj.getBlue();
17
18           pixelObj.setColor(new Color( 255-redValue,
19                                     255-greenValue,
20                                     255-blueValue));
21
22         }
23      }
24   }
```

Line number: ______

Line number: ______

Line number: ______

Java API Documentation for CS026a Midterm Fall 2007

World

Partial Constructor Summary

World()

Constructor that takes no arguments

World(int w, int h)

Constructor that takes a width and height for this world

Turtle

Partial Constructor Summary

Turtle(int x, int y, World world)

Constructor that puts the turtle at position (x,y) in its world, facing "up" (toward the top of the world)

Turtle(World world)

Constructor that puts the turtle at the centre of its world, facing "up" (toward the top of the world)

Partial Method Summary

void backward()

Method to go backward by 100 pixels

void backward(int pixels)

Method to go backward a given number of pixels

void forward()

Method to move the turtle foward 100 pixels

void forward(int pixels)

Method to move the turtle forward the given number of pixels

int getXPos()

Method to get the current x position

int getYPos()

Method to get the current y position

void hide()

Stop showing the turtle; does not affect the pen status

void moveTo(int x, int y)

Method to move to turtle to the given x and y location

void penDown()

Method to set the pen down

void penUp()

Method to lift the pen up

void show()

Make the turtle visible; does not affect the pen status

void turn(int degrees)

Method to turn the turtle the passed degrees (negative to turn left, pos to turn right)

void turnLeft()

Method to turn left 90 degrees

void turnRight()
    Method to turn right 90 degrees
void turnToFace(int x, int y)
    Method to turn towards the given x and y
void turnToFace(SimpleTurtle turtle)
    Method this turtle object to face the parameter turtle

Picture

Constructor Summary

Picture()
    Constructor that takes no arguments
Picture(int width, int height)
    Constructor that takes the width and height
Picture(String fileName)
    Constructor that takes a file name and creates the picture

Partial Method Summary

void explore()
    Method to open a picture explorer on a copy of this simple picture
int getHeight()
    Method to get the height of the picture in pixels
int getWidth()
    Method to get the width of the picture in pixels
Pixel getPixel(int x, int y)
    Method to get a pixel object for the given x and y location
Pixel[] getPixels()
    Method to get a one-dimensional array of Pixels for this simple picture
void repaint()
    Method to force the picture to redraw itself.
void show()
    Method to show the picture in a picture frame
void write(String fileName)
    Store the picture in .jpg format in the file whose name is specified by filename

Pixel

Partial Method Summary

int getBlue()
    Method to get the amount of blue at this pixel
int getGreen()
    Method to get the amount of green at this pixel
int getRed()
    Method to get the amount of red at this pixel
java.awt.Color getColor()
    Method to get a color object that represents the color at this pixel
int getX()

Method to get the x location of this pixel
int getY()
Method to get the y location of this pixel
void setBlue(int value)
Method to set the blue to a new blue value
void setGreen(int value)
Method to set the green to a new green value
void setRed(int value)
Method to set the red to a new red value
void setColor(java.awt.Color newColor)
Method to set the pixel color to the passed in color object.

<u>FileChooser</u>

Partial Method Summary

static String getMediaPath(String s)
Get the full path name for the file specified in s
static String pickAFile()
Let the user choose a file, and return the full file name (path included)

<u>Color </u>(in java.awt)

Partial Constructor Summary

Color(int r, int g, int b)
Constructor creates a Color object with red, green and blue intensities of r, g and b respectively

Partial Method Summary

Color brighter()
Create a new Color object that's a brighter version of this Color
Color darker()
Create a new Color object that's a darker version of this Color
int getBlue()
Return the blue intensity of this Color object
int getGreen()
Return the green intensity of this Color object
int getRed()
Return the red intensity of this Color object
String toString()
Return a string representation of this Color object

Partial Constant Summary

BLACK, black-- A Color object with red, green and blue intensities of 0
BLUE, blue -- A Color object with blue intensity 255, and red and green intensities of 0
GREEN, green-- A Color object with green intensity 255, and red and blue intensities of 0
RED, red -- A Color object with red intensity 255, and green and blue intensities of 0
WHITE, white-- A Color object with red, green and blue intensities of 255

PAGE FOR ROUGH WORK