

*THE UNIVERSITY OF WESTERN ONTARIO  
LONDON, CANADA*

**Computer Science 1026a  
MIDTERM EXAMINATION**

May 28th, 2015

110 minutes

Name: \_\_\_\_\_

**Instructions:**

- Fill in your name above immediately.
- Have your student card out of its case and on the desk.
- For multiple choice and true/false questions, please circle the correct response on the exam paper.
- For short-answer questions, provide your answer in the space provided.
- This is a closed book exam
- If you finish within 15 minutes of the end of the exam, you must wait until the exam ends before leaving so as not to distract those who are still working.
- There is a blank page at the back of this exam for rough work. Additional sheets can be provided upon request. All paper must be returned to your instructor along with your exam.
- No electronic devices are allowed.
- **Please turn off your cell phone.**
- **DO NOT TURN THIS PAGE UNTIL DIRECTED TO DO SO**

<b>Question</b>	<b>Out of</b>	<b>Mark</b>
Part 1	60	
51	10	
52	10	
53	10	
54	3	
55	7	
Total	100	

## PART 1 – Multiple Choice and True/False – 60 marks

*For the following questions, please indicate your answer directly on the exam sheet.*

*Questions are each worth one point unless denoted with [# marks], where # indicates the number of marks for that question.*

*Each question has a single solution. Please clearly indicate only one option.*

Student Number: \_\_\_\_\_

- |   |      |       |
|---|------|-------|
| 1) A bit is a group of 8 bytes  | True | False |
| 2) Java bytecode is a high level language   | True | False |
| 3) Java code is compiled by the Java Virtual Machine  | True | False |
| 4) The statement <code>int x = 1 * 3.14;</code> will compile  | True | False |
| 5) Boolean expressions can only have a value of true or false   | True | False |
| 6) In Java, objects are an example of a primitive type  | True | False |
| 7) The terms declaration and assignment mean the same thing   | True | False |
| 8) Assuming proper Java convention, <code>Math.random();</code> is an example of a class method   | True | False |
| 9) <code>Strings</code> are an example of a primitive type  | True | False |
| 10) Assume the following code is correct:<br><code>Something doodad = someObject.ascd();</code><br>The object method “ <code>ascd()</code> ” returns a <code>Something</code> object. | True | False |

Consider the following code segment when answering questions on this page

```
World w1 = new World(500, 500);
Turtle t1 = new Turtle(w1);
Turtle t2 = new Turtle(w1);
Turtle t3 = t1;
Turtle t4;
t1.forward(50);
t1.forward(50);
t2.forward(50);
```

- 11) The number of **objects** created and or referenced in this code segment is:
- a. 2
  - b. 3**
  - c. 4
  - d. 5
  - e. None of the above
- 12) The number of **reference variables** declared in this code segment is:
- a. 2
  - b. 3
  - c. 4
  - d. 5**
  - e. None of the above
- 13) Which statement best describes what the above code segment does?
- a. Creates 3 turtles and makes all three move forward 100 pixels.
  - b. Creates 2 turtles and moves them each 50 pixels forward.
  - c. Creates 2 turtles, moves 1 forward 100 pixels and the other 50 pixels.**
  - d. Creates 3 turtles and makes all three move forward 50 pixels.
  - e. None of the above statements describes the code segment.
- 14) The 500s in "new World(500, 500);" are examples of parameters **True** False
- 15) t1 and t3 are references to the same object **True** False
- 16) t3 does not contain a reference True **False**
- 17) This code contains an object method **True** False
- 18) forward() is a *class method* True **False**
- 19) t1 refers to an *object* **True** False
- 20) t4 is an example of a *class*. True **False**

Consider the following code segment when answering questions on this page.

```
int[] numbers = {1, 5, 2, 6, 8, 1};
numbers[2] = 0;
System.out.println(numbers[1] + " " + numbers[3] );
```

- 21) Which of the following best describes the declaration in Java?
- Declares an integer variable called `numbers`
  - Declares that the second position of the `numbers` array will be size 0
  - Declares a reference variable to an integer array of size 6
  - Declares 6 reference variables to arrays of sizes 1, 5, 2, 6, 8, and 1
  - Both **a** and **b** are correct
- 22) What text is printed to the screen by the above code segment?
- 5 6
  - 1 2
  - 2 5
  - Array index out of bounds exception
  - Both **a** and **d** are correct
- 23) Which of the following would assign the value 2 to the last element of `numbers`?
- `numbers[numbers.length - 1] = 2;`
  - `numbers[6] = [2];`
  - `numbers[4] = 2;`
  - `numbers[5] = 2;`
  - Both **a** and **d** are correct
- 24) Which of the following statements would correctly declare and create a new integer array called `numbersNew` with the same size as `numbers`?
- `int[] numbersNew = new int[numbers.length-1]`
  - `numbersNew = arrayLength(numbers);`
  - `int numbersNew = Array.length;`
  - `numbersNew == numbers.length;`
  - None of the above

25) Which of the following statements is/are TRUE about RAM?

- a. RAM requires the computer to be powered on to store data
- b. The CPU holds more data than typical RAM
- c. Data in RAM cannot be moved to a register
- d. In a typical computers, RAM is the fastest type of data storage
- e. All of the above are true

26) Which of the following correctly lists the types of memory from slowest to fastest?

- a. Hard disk, RAM, Cache, Registers
- b. Cache, Hard disk, Registers, RAM
- c. ALU, RAM, CPU
- d. Peripheral, Internal, RAM, Registers
- e. Half adder, Full adder, ALU, RAM

27) The following line of code is an example of?

```
int number = 4;
```

- a. A statement
- b. A declaration
- c. An assignment
- d. Both **b** and **c** are correct
- e. All of the above

28) In Java, objects:

- a. Are stored directly in the Full Adder
- b. Are also called *instances* of their Class
- c. Can be modified after they are created
- d. Can be referenced by multiple reference variables
- e. All **but one** of the above are true

29) In Java, classes:

- a. Cannot contain the main method.
- b. Can define the methods and attributes of an object
- c. Require a variable to reference
- d. Do not have any Java conventions regarding their definition
- e. All of the above are true

30) Which of the following statements is/are TRUE about colours in Java?

- a. They are based on the primary colours red, blue and yellow
- b. They are defined in the `.setColor()` class
- c. They are created using hue, saturation and lightness values
- d. They smell bad
- e. None of the above are true

31) How many colours can we represent with a colour described by a single byte for each colour component:

- a. 256
- b.  $2^8$
- c.  $2^{24}$
- d.  $3^8$
- e. Both a and b are correct

32) Consider the following code segment:

```
for ( int i = 1 ; i < 100 ; i-- )
{
    System.out.print(i++ + " ");
}
```

Which of the following best describes what will happen if we run this code?

- a. It will enter an infinite loop, and print numbers forever
- b. It will print all numbers from 0 to 100
- c. It will print all numbers from 100 to 0
- d. It will print all numbers from 1 to 99
- e. None of the above

33) Consider the following code segment:

```
for ( int i = 0 ; i != 99 ; i = i + 2 )
{
    System.out.print(i + " ");
}
```

Which of the following best describes what will happen if we run this code?

- a. It will never enter the loop, and so nothing will print to the screen
- b. It will print all even numbers from 0 to 100
- c. It will print all odd numbers from 0 to 99
- d. It will print all even numbers from 0 to 99
- e. It will enter an infinite loop, and print even numbers forever

Consider the following Java method, which has been added to the `Picture` class when answering questions on this page.

```
public void funPictureMethod()
{
    for (int x=0; x < this.getWidth(); x++)
    {
        for(int y=0; y < this.getHeight(); y++)
        {
            Pixel pix = this.getPixel(x,y);
            pix.setColor( pix.getColor().darker() );
        }
    }
}
```

- 34) Which of the following statements is/are TRUE about the method?
- The method will turn all pixels in the image white
  - The method will turn all pixels in the image darker**
  - The method will not do anything to the image
  - The method contains a logical error and will not run properly
  - None of the above are true
- 35) What is true about the method `funPictureMethod`:
- The method returns something
  - We create a reference variable for a `Color` object
  - The `Picture` class now has a *class method* called `funPictureMethod`
  - The method has nested loops**
  - All of the above are true
- 36) In the above method, “`this`” is a reference to:
- A `Picture` object we have created and named `this`
  - Any `Picture` object on which the method is invoked**
  - Every `Pixel` object in the image
  - A class
- 37) In the above method, in what order are the nested for loops modifying pixels?
- row by row
  - column by column**
  - x by y
  - y by x
  - Diagonally

Consider the following code segment when answering questions on this page

```
for (int i = 1; i < 4; i++){
    System.out.println("+");
    for (int j = i; j > 0; j--){
        System.out.print("j");
    }
    System.out.print("+");
}
```

38) What will be printed to the screen by the above code?

- a. +1+21+321+
- b. +1+2+3+
- c. + (ONLY OUT OF 1, NOT 4),  
1++ I messed up this question  
21++  
321+
- d. +j++jj++jjj+
- e. This code segment will produce an error.

39) Which of the following statements about the above code is true?

- a. It contains a nested loop
- b. It will result in a total of 3 two calls to the `System.out.print` method
- c. It does not contain a complete String expression
- d. None of the above

40) How many times will the outer loop repeat its body of code?

- a. 0
- b. 2
- c. 3
- d. 4
- e. None of the above

41) In total, how many times will the boolean expression `j > 0` be evaluated?

- a. 0
- b. 2
- c. 3
- d. 4
- e. None of the above

42) Which of the following is a Boolean **operator**?

- a. true
- b. false
- c.  $a \leq 8$
- d.  $4 > 5$
- e. !

43) Which of the following is/are compiler errors?

- a. Forgetting a semicolon
- b. Failing to follow naming conventions
- c. Mismatched brackets
- d. Array index out of bounds
- e. Both a and c are correct

44) Which of the following **are** bugs?

- a. Forgetting a semicolon
- b. Failing to follow naming conventions
- c. Not defining a variable before use
- d. Mismatched brackets
- e. Array index out of bounds

If they selected more than one, mark correct if one was right

45) Consider the statement below. Assume  $x$  and  $y$  are declared as type double

```
x = (int)x*y;
```

What is being cast by the casting operator?

- a.  $x$
- b.  $y$
- c.  $x*y$
- d. `int`
- e. None of the above

46) Consider the statement below. Assume  $x$  and  $y$  are declared as type double

```
x = (int)x + x*y;
```

What is being cast by the casting operator?

- a.  $x$
- b.  $x*y$
- c.  $x + x*y$
- d. `int`
- e. None of the above

47) Consider the following code segment:

```
double x = 1.5;
double y = 1;
x = (int)x + x*y;
```

What data type is stored in x in the last line?

- a. double
- b. integer
- c. intDouble
- d. N/A: this code is nonsense
- e. None of the above

48) Consider the following program, saved in the file Hello.java

```
public class Hello
{
    public static void main (String[] args)
    {
        System.out.println("Hello!!");
    }
}
```

Will the above program compile and run?

- a. Yes
- b. It will compile but not run
- c. No, it will not compile

49) The class name must match the file name True False  
(class "Hello" and file "Hello.java")

50) String[] args means the same as String args[] True False

## PART 2 – Written Answer – 40 marks

For the following questions, please write your answers in the space provided.

- 51) **[10 marks]** For the following Java expressions, indicate its value (the result of evaluating the expression) and its type. If it is not a valid expression, indicate that in the Value column and leave the Type column empty.

	Value	Type
<code>( 4 + 6 ) % 4</code>	2	Int
<code>6 &gt;= 5</code>	TRUE	Boolean
<code>'z' != 1</code>	TRUE	Boolean
<code>'a' = 'A'</code>	NONSENSE	NONSENSE
<code>(3 &gt; 3)    (4 &gt; 3)</code>	TRUE	Boolean
<code>5 * 3.0 + (4 / 3) + 2</code>	18.0	Double
<code>"hello " + \" + "world"</code>	NONSENSE	NONSENSE
<code>TRUE &amp; TRUE &amp; TRUE &amp; TRUE &amp; FALSE</code>	FALSE	BOOLEAN
<code>"James Rox" + 4 + "ever"</code>	James Rox4ever	String
<code>"CS" + "1026" + "a"</code>	CS1026a	String

52) [5 marks] Write an object method to be added to Turtle class in the Turtle.java file called drawBoxRow (header provided below)

The method will draw a row of boxes:

- Assume the turtle starts by facing the top of the world it is in
- The box's sides must be `sideLength`
- There must be `sideLength` pixels between each box
  - i.e. if `sideLength` is 50, it will have 50 pixels between boxes
- The method must draw `numBoxes` boxes
  - i.e. if `numBoxes` is 5, it will draw 5 boxes
- Comments are not required, but may be used to explain your approach. They will be considered when awarding part marks.
- You must use multiple nested for loops

```
public void drawBoxRow(int sideLength, int numBoxes)
{
    //write your code under here

    for(int i = 0; i < numBoxes; i++)
    {
        this.turnRight();
        this.forward(sideLength);
        this.turnRight();
        this.forward(sideLength);
        this.turnRight();
        this.forward(sideLength);
        this.turnRight();
        this.forward(sideLength);
        this.penUp();
        this.forward(sideLength * 2);
        this.penDown();
    }

    //They could also have put the box drawing in a for loop

}
```

53) [10 marks] For this question you are to write an object method for a picture object. The method is to perform the following:

- Create a new picture object called `newPic` twice the size of the original picture
- Iterate through each pixel in the original picture and copy it into `newPic` such that the new picture will be exactly twice the size of the original.
  - o NOTE/HINT: `newPic` will have 4x the number of pixels as the original!
- Methods you might need to use:
  - o `getPixel(x,y)`
  - o `getColor()`
  - o `setColor()`
  - o `getWidth()`
  - o `getHeight()`

```
public Picture makeDoubleSizePicture()
{
    Picture newPic = new Picture(this.getWidth() * 2,
    this.getHeight() * 2); THEY MUST HAVE THIS
    LINE PERFECT, OTHERWISE THEY GET DINGED
    Pixel pix;
    Color col;
    for(int x = 0 ; x < this.getWidth() ; x ++)
    {
        for(int y = 0 ; y < this.getHeight() ; y++)
        {
            pix = this.getPixel(x,y);
            col = pix.getColor();
            newPic.getPixel(2*x,2*y).setColor(col);
            newPic.getPixel(2*x+1,2*y).setColor(col);
            newPic.getPixel(2*x,2*y+1).setColor(col);
            newPic.getPixel(2*x+1,2*y+1).setColor(col);
        }
    }

    return newPic; //returns the NEW picture
}
```

54)[3 marks] For the following code segment, rewrite the segment and convert the for loop to a while loop. The functionality of the code segment must not be altered by your modifications. You may alter the variable names if it makes it easier for you.

```
int nums[] = new int[20];
for ( int i = 0; i <= nums.length; i++)
{
    nums[i] = i;
}
```

```
int nums[] = new int[20];
int i = 0;
while(i <= nums.length)
{
    nums[i] = i;
    i++;
}
```

They can keep the bug when re-writing or remove it when re-writing. Either way, I'm happy

55) [1 marks] What is the bug in this code (don't explain why, just write the bug)?

`i <= nums.length` (should be `i < nums.length` OR `i <= nums.length-1`) OR `nums[]`

56) [1 marks] Will the code *compile* (write Yes or No)? YES OR NO if `nums[]`

57) **[10 marks]** Write a code segment that will generate the output as seen below using for loop(s) nested within other for loop(s). You have an array of letters, as declared below, which must be used in the printing of your solution for full marks. Think carefully about your approach, and consider breaking down the process of creating each line into separate steps. The solution must be simple/minimal, and use the `letters` array as defined to get full marks.

```
e
cec
aceca
racecar

// given
char[] letters = { 'r', 'a', 'c', 'e' };

// put your code segment here:

int len = letters.length;

for(int i=0; i< len; i++)
{
    for(int j= len-1-i; j < len; j++)
    {
        System.out.print( letters[j] );
    }
    for(int j= len-2; j >= len-1-i; j--)
    {
        System.out.print( letters[j] );
    }
    System.out.println();
}
```

## Turtle Methods

	Method Functionality	Method Syntax	Notes
Creating a Turtle	Constructors	Turtle(World world)	Default: starts facing north and in center of world
		Turtle(int x, int y, World world)	Starts at position (x,y) (x starts at 0 on the left, y starts at 0 at the top of the window)
Moving a Turtle	Move forward	void forward()  void forward(int pixels)	Move forward by 100 pixels  Move forward by given number of pixels
	Move backward	void backward()  void backward(int pixels)	Move backward by 100 pixels  Move backward by given number of pixels
	Move to a particular location	void moveTo(int x, int y)	Move to location (x,y)
Turning a Turtle	Turn right	void turnRight()	Default: turn right by 90 degrees
	Turn left	void turnLeft()	Default: turn left by 90 degrees
	Turn right or left	void turn(int degrees)	Turn right if degrees is positive, left if degrees is negative
	Turn to face another turtle	void turnToFace(Turtle turtle)	Turn to face turtle
	Turn to face a specific point	void turnToFace(int x, int y)	Turn to face point (x,y)
	Turn to face a specific heading	void setHeading(int degree)	Turns the turtle to face a degree heading between 0 and 359. 0 faces the top of the World and degrees increase in the clockwise direction
Turtle Color	Change turtle color	void setColor(Color color)	Also sets the pen color
	Change turtle body color	void setBodyColor(Color color)	
	Change turtle shell color	void setShellColor(Color color)	
	Get turtle color	Color getColor()	
	Get turtle body color	Color getBodyColor()	
	Get turtle shell color	Color getShellColor()	
Turtle Size	Change turtle's width or height	void setWidth(int width) void setHeight(int height)	

	Get turtle's width or height	int getWidth() int getHeight()	
Turtle Name	Set turtle's name	void setName(String name)	
	Get turtle's name	String getName()	
Turtle Position	Get turtle's position	int getXPos() int getYPos()	
Turtle Visibility	Hide turtle	void hide()	Does not affect pen status
	Show turtle	void show()	
Turtle Information	Get turtle information	String toString()	
The Pen	Pick up pen	void penUp()	Default: the pen is down and leaves a trail
	Put pen down	void penDown()	
	Check if pen is down	boolean isPenDown()	
	Change width of pen trail	void setPenWidth(int width)	
	Set / change pen color	void setPenColor(...)	
	Get pen color	Color getPenColor()	
	Get pen width	int getPenWidth()	

*This page is intentionally left blank for rough work*

*This page is intentionally left blank for rough work*