



First name: _____ Last name: _____	AYA Tues 09:00am-10:50 AYB Tues 11:00am-12:50 AYC Tues 01:00pm-02:50 AYD Tues 03:00pm-04:50 AYE Tues 05:00pm-06:50 AYF Tues 07:00pm-08:50 AYG Wed 09:00am-10:50 AYH Wed 11:00am-12:50 AYK Wed 01:00pm-02:50 AYI Wed 03:00pm-04:50 AYJ Wed 05:00pm-06:50
NetID: _ _ _ _ _ @ illinois.edu (Please write legibly)	
Discussion Section: AY _ (We will return your exam manuscript to you in this section)	

- This is a closed book and closed notes exam. No electronic aids are allowed. You must turn this exam booklet in before leaving.
- You should have 9 sheets total including one scratch sheet. The last sheet is scratch paper; you may detach it while taking the exam, but must turn it in with the exam when you leave.
- You should not write any additional methods for any problem in this exam; you only complete the main() method for that problem.
- The TAs will not answer any technical questions, they will only clarify non-technical English words for non-native speakers. If you believe a question is ambiguous or contains an error, write down your assumptions and answer accordingly.
- You may assume the user will only provide acceptable input to the program.
- When you write code, you may use a shorthand for System.out and TextIO input and output methods provided it is obvious to the graders which method you are using. For example it is acceptable to use Sopl in place of System.out.println and to use Sopt in place of System.out.print.
- For full marks correct syntax is required: Ensure all statements include a semicolon and the correct use of upper/lower case, single quotes and double quotes. However a syntactically correct but flawed program will earn a low score i.e. semantics and correctness are more important than minor typos and syntax errors.

Problem	Points	Score	Grader
1	9		
2	10		
3	14		
4	15		
5	15		
6	12		
Total	75		

1. Type Analysis and Machine Code – 9 points (3+3+3 points)

For each of the code snippets below, there are **zero**, **one** or **more** type errors that the compiler would catch. Circle zero, one or two for the number of type errors and explain what the type error(s) are. Indicate what types and operators are involved in the erroneous part of the expression. Circle zero if there is no type error. Examples:

Example	Response
<code>int a=4+"hi";</code>	1: A string value is being assigned to an integer variable.
<code>true==3</code>	1: An integer value is being compared with a boolean.
<code>int c=6*3;</code>	0

(a)

```
String text = TextIO.getln();
int result = text.length() > (int)(6*Math.random());
```

Circle the correct response (0,1, or 2). If one or two then specify the types and the operation(s) involved that cause the type error.

Number of Type Error(s)? 0 / 1 / 2	If 1 or 2, explain:
---------------------------------------	---------------------

(b)

```
int x = TextIO.getlnInt() * 2;
String result = "ABC" + x;
```

Number of Type Error(s)? 0 / 1 / 2	If 1 or 2, explain:
---------------------------------------	---------------------

(c) Consider the following assembly code for a simple processor:

```
LOAD 0 -> R0          # R0 is now zero.
LOAD 1 -> R1          # R1 is now one.

ADD 1 + R0 -> R0      # R0 is incremented by one.
ADD R1 + R1 -> R1     # R1 is doubled.
SUB R5 - R1 -> R4     # R4 holds the result of R5-R1 (and condition codes are updated).
BR.ZP -4              # Will jump back to the first add instruction if the zero or positive condition code is set.
```

What are the final values of R0 and R1 after the above code completes?

Initial value of R5	Final value of R0	Final value of R1
0		
2		
9		

2. Execution Analysis – 10 points

(a) Read the code below then answer the four questions on the right.

```
int a = 0, b = 0, c = -1;
boolean p = true, q = true;
while(b<50) {
    c = a;
    a = TextIO.getlnInt();
    p = b == c;
    if(p) TextIO.putln(b);
    if(p && b<10) q= false;
    b++;
}
```

What is the smallest possible value printed by this code, irrespective of user's input?

Choose one variable that best illustrates the following concepts.

One-way flag _____

Follower/Previous value _____

Stepper/Iteration counter _____

(b)

Identify the kind of error in the code below.

```
1 while(Math.random()<0.5) {
2     int c = (int)(Math.random() * 42);
3 }
4 TextIO.putln(c);
```

Type Runtime Segmentation Scope error (circle one)

(c)

Carefully analyze the following code. What will it print when executed? For partial credit show your working.

```
for (int a = 10; a > 1; a = a-1) {
    TextIO.put(a);
    TextIO.put(" ");
    a = a/2;
}
```

Output : _____

(d)

Write one value for x that will cause the last subexpression, $z < 10$, to be evaluated?

```
int x = ...not shown...
int z = ...not shown...
if ( (x%2 != 1) || (x<8) || (z < 10) ) TextIO.putln();
```

Your answer: x= _____

3. Pizza Service – 14 Points

PerfectPizza provides custom Pizza recommendations based on the customer's system. Complete the program below to print one of “C”, “E”, “P”, or “S”, corresponding to Cheese, Everything, Pepperoni or Surprise types of pizza respectively, using the following rules:

- If none of the other rules below are satisfied, then recommend a **Cheese** pizza.
- Express orders always get the **Surprise** pizza.
- Non-express orders with 5 or more people who are watching a movie or partying, get **Pepperoni**.
- Non-express orders with 3 people studying get the **Everything** pizza.

Read the following carefully: Complete the given code below. Use only if,else if,or else statements and **four** output statements – one to print each output. That is, **do not** print out “P” in two different places in the code, or assign it to a variable – your boolean expressions should be such that you do not have multiple statements in your code which print out the same letter. Do not create any additional variables or methods.

```
public class PerfectPizza {
    public static void main (String[] args){
        TextIO.putln("Purpose:Movie? Party? Studying?");
        TextIO.putln("Type M, P, or S and press return");
        char purpose = TextIO.getlnChar(); // assume user types M P or S

        TextIO.putln("Express order?")
        boolean express = TextIO.getlnBoolean();

        TextIO.putln("Number of people?");
        int people = TextIO.getlnInt(); // assume user types a positive integer
```

```
// Please continue overleaf or on the spare page if you need more space.
    } // end of main method
} // end of class
```

4. Text Processing (With a thermal analysis of your keyboard I found the password) – 15 Points

Complete the following program to process lines from a text file, which contains a transcript of a conversation between four pirates (Lawrence, Cinda, Lenny and Rob). Every line in the file starts with the speaker's two initials, a colon, followed by a transcript of their speech-

```
LP:We should take the ship at night...
LA:But security will be tighter at the harbor.
CH:And we have the element of surprise!
RR:Our monkey will take care of that.
```

Complete the following to print out all of Lawrence's lines (LA) where the previous line is by Lenny Pitt (LP) and includes the text "gold" or "ship" (in upper/lower or mixed case). Stop reading lines from the file when 32 of Lawrence's lines have been printed (you can assume the file has at least this many examples). You can print out the entire line including "LA:"

The following string methods may be useful: `.equals(string)`, `.length()`, `.charAt(int)`, `.toArray()`, `.indexOf(string)`, `.toUpperCase()`. `TextIO.getln()` can be used to read the next line from the file. `TextIO.eof()` is unnecessary for this problem.

```
public class MyReader {
    public static void main(String[] args) {
        TextIO.readFile("Transcript.txt"); // start reading the file
```

```
    } // end of main method
} // end of class
```

5. Perfect code – 15 Points

For problems i-iv, a user has reported a problem. Show where the error(s) are and fix the code appropriately. There may be more than one error per problem.

i) Support email: "It always says I passed!"

Fix the code so that it only prints when the score is greater than ten

```
boolean isOK = score > 9;
if( isOK = true) TextIO.putln("You got more than ten correct!");
```

ii) Support email: "I get ten correct but the program seems to just stop and not do anything!"

The variable *count* is initially a positive integer. Fix the code so that all three statements are executed by the while loop and the number of stars printed equals the initial value of *count*.

```
while(count>1)    passed++; count --; TextIO.put( "*" );
```

iii) Support email: "It doesn't work properly when I just press return"

Fix the code so that an empty response is treated the same as entering "N"

```
String input = TextIO.getln();
boolean ok = input.charAt(0) == 'Y';
```

iv) Support email: "It never finds the spoon!"

Fix the code so that "Found" is printed (without the quotes) if the line is exactly spoon.

```
String line = TextIO.getln();
if( line == "spoon"); TextIO.putln("Found");
```

v) Integer variables *a* and *c* are initially zero. Rewrite the following *while* loops to use *for* loops. What will be the final value of *c*?

```
while( a<5 ) {
    a++;
    int b = 0;
    while( b < a && (a%2 ==0) )
        b++;
    c = c + b;
}
```

Final value of *c*: _____

6. Secret (Last Question) – 12 points

There's a *secret* encoded in this message: "as one tic? or the tent"

For the above message your program will print "secret".

For the message "bill!no v due! !", your program will print "love"

Complete the program below to print the last letter of every word in lowercase. A word is a sequence of letters. A letter is *defined* as *only* the characters 'a' to 'z'. All words, except the first one, are preceded by one or more non-letters (?!space,) etc (this list is incomplete). The line may end with a letter or non-letter character.

Cryptic hints: String methods `charAt(integer)` and `length()`, a loop and a boolean variable or two might be useful. The expression like `'a' <= ch && ch <= 'z'` might be useful. For full credit implement the solution using a single loop.

```
public class Decode {
    public static void main(String[] args) {

        TextIO.putln("Enter the entire message to decode");

        String m = TextIO.getln().toLowerCase(); // now m is all lowercase
```

```
    } // end of main method
} // end of class LAST QUESTION. GO BACK AND CHECK YOUR WORK.
```

Empty Page

NetID: _ _ _ _ _

Scratch paper - you may detach it while taking the exam, but must turn it in with the exam when you leave.