

Birkbeck
(University of London)
Software and Programming 1
In-class Test 2
26 Mar 2020

Student Name _____
Student Number _____

Answer ALL Questions

1. What output is produced when the following Java program fragment is executed? You should show your workings.

```
int p = 3;
for (int n = 2; p < 50; n += 3)
    p += n;
System.out.println(p);
```

(6 marks)

Answer: 60 (there is only one print out because the `println` statement is outside the loop)

Workings:

before			after	
p	n	p < 50	s	n
3	2	3 < 50 is true	5	5
5	5	5 < 50 is true	10	8
10	8	10 < 50 is true	18	11
18	11	18 < 50 is true	29	14
29	14	29 < 50 is true	43	17
43	17	43 < 50 is true	60	20
60	20	60 < 50 is false		

2. Let `title` be a variable of type `String`. Write a Java expression (of type `String`) that is evaluated to `title` enclosed in "[" and "]" if `title` is no longer than 10 characters the first 7 characters of `title` enclosed in "[" and "...]" otherwise. For example, if `title` is "hello", then the value of the expression should be "[hello]"; if `title` is "greetings, humans", then the value of the expression should be "[greetin...]".

Assume `title` is not null. Your expression, however, should never throw an `IndexOutOfBoundsException`: recall that method

```
String substring(int beginIndex, int endIndex)
```

of class `String` throws an `IndexOutOfBoundsException` if the `beginIndex` is negative, or `endIndex` is larger than the length of this `String` object, or `beginIndex` is larger than `endIndex`.

An expression *cannot* contain `if` or `return` statements; instead, one should be able to place it as *expr* in the following Java code:

```
public class T { public static void main(String[] args) { System.out.println(expr); } }
```

Answer: `"[" + (title.length() <= 10`
`? title`
`: title.substring(0,7) + "...") + "]"`

3. What output is produced when the following Java program fragment is executed
- ```
if (d >= 0) if (d == 1) System.out.print("U"); else System.out.print("Z");
System.out.print("N");
```

after each of the following declarations:

- (a) `int d = 0;`
- (b) `int d = 1;`
- (c) `int d = -1;`

You should show your workings.

(5 marks)

**Answer:** (a) ZN (b) UN (c) N  
the `else` belongs to the second `if`  
the last `print` is executed every time because it is not part of the `ifs`

4. A *run* is a sequence of adjacent repeated values. Implement a method

```
public static void printRuns(String[] v)
```

that prints the array with each run of length greater than 1 marked by including it in parentheses.

For example, on the input array

```
a b e e c a a a b d c b b b b c f e e c a
```

the method should produce the following output:

```
a b (2 e) c (3 a) b d c (4 b) c f (2 e) c a
```

(the 2 es, the 3 as, 4 bs and 2 es all form runs of length greater than 1, and the number after the bracket specifies the length of the run). **(22 marks)**

**Answer:**

```
public static void printRuns(String[] v) {
 if (v.length == 0)
 return;
 String last = v[0];
 int runLength = 1;
 for (int i = 1; i < v.length; i++) {
 if (last.equals(v[i]))
 runLength++;
 else {
 if (runLength > 1)
 System.out.print("(" + runLength + " " + last + ") ");
 else
 System.out.print(last + " ");
 last = v[i];
 runLength = 1;
 }
 }
 if (runLength > 1)
 System.out.print("(" + runLength + " " + last + ")\n");
 else
 System.out.print(last + "\n");
}
```

5. Implement a method

```
public static double[] product(double[] v1, double[] v2)
```

that, given two arrays, *v1* and *v2*, of floating-point numbers, returns a *new* array containing the component-wise product of vectors *v1* and *v2*, that is, an array whose *i*th element is  $v1[i] * v2[i]$ .

For example, if *v1* is { 5, 2, 3 } and *v2* is { -1, 2, 0 }, then the method should return { -5, 4, 0 }.

(10 marks)

**Answer:**

```
public static double[] product(double[] v1, double[] v2) {
 double[] result = new double[v1.length];
 for (int i = 0; i < result.length; i++)
 result[i] = v1[i] * v2[i];
 return result;
}
```

6. What output is produced when the following Java program is executed?

```
public class E20 {
 public static void main(String args[]) {
 int[] a = { 1, 2, 3, 1 };
 System.out.println(a[h(a, 1)] + 1 == a[h(a, 2)] ? "yes" : "no");
 }
 public static int h(int[] v, int i) {
 return v[v[i]];
 }
}
```

You should show your workings.

(10 marks)

**Answer:**

yes

**Workings:** When the method h is called in h(a, 1), v in h refers to the array { 1, 2, 3, 1 } and so, we get:

$v[v[i]] \rightarrow v[v[1]] \rightarrow v[2] \rightarrow 3$  and so,  
return v[v[i]]; returns 3

When the method h is called in h(a, 2), v in h refers to the array { 1, 2, 3, 1 } and so, we get:

$v[v[i]] \rightarrow v[v[2]] \rightarrow v[3] \rightarrow 1$  and so,  
return v[v[i]]; returns 1

As  $a[3] + 1$  is  $1 + 1$  and  $a[1]$  is 2, the method main prints "yes".

7. Suppose you have declared a class `Question` as follows:

```
public class Question {
 public int getPoints(String answer) { return 0; }
}
```

- (a) Write a subclass `SimpleTextQuestion` of class `Question`. The class should have two instance variables: `correctAnswer` of type `String` and `points` of type `int`. It should have a single constructor taking the two parameters and storing them in the instance variables. Implement instance methods `getPoints(String answer)` (that compares the provided answer with the correct answer and returns points if they match and 0 otherwise) and `getCorrectAnswer()` (that returns the stored correct answer). (7 marks)
- (b) Write a subclass `MultipleChoiceQuestion` of class `Question`. The class should have three instance variables: `options` of type `String[]`, and `correctAnswerIndex` and `points` of type `int`. It should have a single constructor taking the three parameters and storing them in the instance variables. Implement methods `getPoints` and `getCorrectAnswer` (see Item (a) for an explanation of the two methods). (7 marks)
- (c) Override method `toString()` in the two classes, `SimpleTextQuestion` and `MultipleChoiceQuestion`, in such a way that they each return a suitably constructed string representation of the instances. (7 marks)
- (d) Write a class `Questionnaire`, whose instances can contain up to 10 Questions. Implement the following instance methods:
- `void addQuestion(Question question)` adds the question to the questionnaire (if the capacity is exceeded, then the method should do nothing);
  - `int getCount()` returns the number of questions in the questionnaire.
- (7 marks)
- (e) Implement the following instance method in class `Questionnaire`:
- `int getPoints(String[] answers)` returns the total number of points for the answers provided (assume that the length of answers coincides with the number of questions in the questionnaire).

(12 marks)

**Answer:** (a)

```
public class SimpleTextQuestion extends Question {
 private String correctAnswer;
 private int points;

 public SimpleTextQuestion(String correctAnswer, int points) {
 this.correctAnswer = correctAnswer;
 this.points = points;
 }

 public int getPoints(String answer) {
 // or: return answer.equals(getCorrectAnswer()) ? points : 0;
 return answer.equals(correctAnswer) ? points : 0;
 }

 public String getCorrectAnswer() {
 return correctAnswer;
 }
}
```

(b)

```
public class MultipleChoiceQuestion extends Question {
 private String[] options;
 private int correctAnswerIndex;
 private int points;

 public MultipleChoiceQuestion(String[] options, int correctAnswerIndex,
int points) {
 this.options = options;
 this.correctAnswerIndex = correctAnswerIndex;
 this.points = points;
 }

 public int getPoints(String answer) {
 // or: return answer.equals(getCorrectAnswer()) ? points : 0;
 return answer.equals(options[correctAnswerIndex]) ? points : 0;
 }

 public String getCorrectAnswer() {
 return options[correctAnswerIndex];
 }
}
```

(c)

```
public String toString() {
 return "simple text question, correct answer: " + correctAnswer
 + ", points: " + points;
}
```

```
public String toString() {
 return "multiple choice question, options: "
 + java.util.Arrays.toString(options) + ", correct answer index: "
 + correctAnswerIndex + ", points: " + points; }
}
```

(d)

```
public class Questionnaire {
 private Question[] questions = new Question[10];
 private int count = 0;

 public void addQuestion(Question question) {
 if (count < questions.length) {
 questions[count] = question;
 count++;
 }
 }

 public int getCount() {
 return count;
 }
}
```

(e)

```
public int getPoints(String[] answers) {
 int total = 0;
 for (int i = 0; i < answers.length; i++)
 total += questions[i].getPoints(answers[i]);
 return total;
}
```