Birkbeck
(University of London)
Software and Programming 1
In-class Test 1.1
13 Feb 2020

Student Name _____
Student Number _____

Answer all questions

1. Consider the following sequence of Java statements:

```
int i = 2;
int k = 13 % i;
i = 20 * -k + i * k * 42 - i;
k = 32 / i + i * 2;
```

What is the value of `k` after these statements are executed? Show your workings. (**7** marks)

Answer: `124`

Workings: (a) `k = 1`
(b) `i = 20 * (-1) + 2 * 1 * 42 - 2 = 62`
(c) `k = 32 / 62 + 62 * 2 = 124`

2. Given variables `speed` of type `int` and `direction` of type `String`, write an expression of type `boolean`, which is evaluated to `true` if the `speed` is above `21` and `direction` is `"N"`, `"NE"` or `"NW"`, and to `false` otherwise. (**7** marks)

Answer:
```
speed > 21 && (direction.equals("N") || direction.equals("NE") ||
      direction.equals("NW"))
```
sp1-02-20.pdf / p. 22 and p. 26

note that the brackets around `||` with arguments are required — see sp1-03-20.pdf / p. 27

3. How many iterations do the following loops carry out? Assume that `i` is not changed in the loop body.

   (a) `for (int i = 0; i <= 100; i += 3) ...`

   (b) `for (int i = 100; i > 0; i-= 2) ...`

(**4** marks)

Answer: (a) 34 (b) 50

4. Which of the following are valid Java identifiers (i.e., possible names of variables/methods)? Note that `0` is a digit (zero) and `O` is a letter.

   (a) `string`
   (b) `repeat`
   (c) `Byte`
   (d) `foreach`
   (e) `len`
   (f) `0xFF`
   (g) `xFF`

(**7** marks)

Answer: (a), (b), (c), (d), (e), (g) sp1-01-20.pdf / p. 21

5. Identify and explain five compile-time errors in the following Java code:

```
1  import java.util.Scanner;
2  public class foo_bar {
3      public static main(String[] args) {
4          double sum = 0;
5          int count = 0;
6          Scanner s = new Scanner(System.in);
7          for (boolean DONE = false; !DONE; DONE = DONE)
8              System.out.println("Input a number: ");
9              double num = s.nextDouble();
10             if (num <> 0) {
11                 sum += num;
12                 count++;
13             }
14             else
15                 DONE = True;
16         }
17         System.out.println("Average: " + num / count);
18     }
19 }
```

How would you correct the errors you have found (with as few changes as possible)?

(**10** marks)

Answer:

line 3: missing return type `void`
line 7: missing {
line 10: replace <> with `!=`
line 15: replace `True` with `true`
line 17: `num` is out of scope, replace with `sum`

6. Implement a method `getState` to determine whether water is liquid, solid (ice) or gaseous at the sea level given the temperature value and the string `"C"` for Celsius and `"F"` for Fahrenheit. The method should take one argument of type `double` and one argument of type `String` and return a `String` according to the following table:

Celsius	Fahrenheit	
below 0	below 32	solid
0–100	32–212	liquid
above 100	above 212	gaseous

If the second argument is different from `"C"` and `"F"`, then the method should return the empty `String`.

(**15** marks)

Answer:

```
public static String getState(double t, String s) {
   if (s.equals("C")) {
      if (t < 0)
         return "solid";
      if (t < 100)
         return "liquid";
      return "gaseous";
   }
   if (s.equals("F")) {
      if (t < 32)
         return "solid";
      if (t < 212)
         return "liquid";
      return "gaseous";
   }
   return "";
}
public static String getState2(double t, String s) {
   if (!s.equals("C") && !s.equals("F"))
      return "";
   if (s.equals("C") && t < 0 || s.equals("F") && t < 32)
      return "solid";
   if (s.equals("C") && t < 100 || s.equals("F") && t < 212)
      return "liquid";
   return "gaseous";
}
```


7. What is printed as a result of executing the following fragment of code?

```
int i = 2;
int k = i + 5;
while (k < 16) {
   i = i + 2;
   System.out.println(k - 5);
   k = i + 5;
}
```

Show your workings. (**10** marks)

Answer:

```
2
4
6
8
10
```

Workings:

i	k	k < 16	new i	printout k - 5	new k
2	7	true	4	2	9
4	9	true	6	4	11
6	11	true	8	6	13
8	13	true	10	8	15
10	15	true	12	10	17
12	17	false			

8. What are the type and the value of the following expression

```
scale.equals("F") && (t - 32) * 5 / 9.0 > 100 ||
scale.equals("C") && t > 100 ? "steam" : "no steam"
```

with the following declarations: `String scale = "C"; int t = 92;`? (**5** marks)

Answer:

`String` and `"no steam"` (because the first argument of the first logical AND is `false`, the first argument of the second logical AND is `true`, but `t > 100` is `false` — see sp1-03-20.pdf / p. 27 on operation precedence)

9. Implement a method that returns `true` if its argument of type `int[]` (array of integers) is a sequence of numbers `0`, `1` and `3` that contains at least one occurrence of `0`.
For example, it should return `false` on {`1, -1, 1`}, {`1, 2`}, {}, {`1, 0, 2`}, and {`3, 1`}, but `true` on {`1, 0, 3`} and {`0`}. (**15** marks)

Answer:

```
public static boolean match(int[] s) {
   boolean found0 = false;
   for (int i = 0; i < s.length; i++) {
      if (s[i] == 0)
         found0 = true;
      if (s[i] != 0 && s[i] != 1 && s[i] != 3)
         return false;
   }
   return found0;
}
public static boolean match2(int[] s) {
   boolean found0 = false;
   for (int e : s) {
      if (e == 0)
         found0 = true;
      if (e != 0 && e != 1 && e != 3)
         return false;
   }
   return found0;
}
```


10. (*a*) Transform the `for` loop in the following fragment of code into a `while` loop.

```
int points = 0;
for (int d = 0; d < g.length/2; d++)
   if (g[d*2 + 1] > g[d*2])
      points += 3;
   else if (g[d*2 + 1] < g[d*2])
      points += 1;
System.out.println("points: " + points);
```

(*b*) Suppose that `g` is declared as follows: `int[] g = { 0, 2, 3, 1, 1, 1 };`. What is printed out as a result of executing this fragment of code? Show your workings. (**20** marks)

Answer:

(a)

```
int points = 0;
int d = 0; // first argument of the for loop
while (d < g.length/2) { // second argument of the for loop turns into
   if (g[d*2 + 1] > g[d*2])
      points += 3;
   else if (g[d*2 + 1] < g[d*2])
      points += 1;
   d++;      // do not forget the third argument of the for loop
}         // and curly brackets
System.out.println("points: " + points);
```

(*b*) The output is: `points: 4`

