University of Glasgow

Wednesday, 6 May 2015
2.00pm – 4.00pm
(Duration: 2 hours)

DEGREES OF MSc in Information Technology, MSc in Software Development

PROGRAMMING

(Answer all 5 questions.)

This examination paper is worth a total of 75 marks

This page is intentionally left blank

Continued overleaf/

1. Consider the following switch statement:

```
switch(month) {

case 2:
case 4:
case 6:
   System.out.println("Thai Food");

case 1:
case 3:
case 5:
   System.out.println("Indian food");
   break;

default:
   System.out.println("British food");

}
```

(a) Write what the program outputs to the console when `month=4` and explain why. [2]

(b) Write what the program outputs to the console when `month=5` and explain why. [2]

(c) Write what the program outputs to the console when `month=9` and explain why. [2]

Consider now the following statements (`a`, `b` and `val` are all of type `int`):

```
val = (a<b) ? a:b;
System.out.println(val);
```

(d) Write what the program outputs to the console when `a=3` and `b=2` and explain why. [2]

Modify the statements above as follows:

```
val = (a=b) ? a:b;
System.out.println(val);
```

(e) Is the syntax of the modified statements correct? Explain your answer. [2]

2. Write a method `getMultiples` that takes as input three positive integers (`start`, `stop` and `divisor`), loops over the integers between `start` and `stop` (included) and, each time an integer in this range can be divided exactly by `divisor`, outputs a message to the console according to the following format:

```
Integer x can be divided exactly by y: x/y=z
```

where `x`, `y` and `z` are integers (e.g., if `start=1`, `stop=100` and `divisor=7`, the first line that the method will output is “`Integer 7 can be divided exactly by 7: 7/7=1`”, the second line that the method will write is “`Integer 14 can be divided exactly by 7: 14/7=2`”, and so on).

Once all the numbers between `start` and `stop` (included) have been considered, the method should output the following sentence: “`Between a and b there are c numbers that can be divided exactly by d`”, where `a`, `b`, `c` and `d` are integers (e.g., if `start=1`, `stop=100` and `divisor=7`, the sentence will be “`Between 1 and 100 there are 14 numbers that can be divided exactly by 7`”).

[10]

3. Consider the following initialisations:

```
String a = "1";
char b = '2';
int c = 10;
double d = 4.0;
```

Explain what a program outputs to the console and why when the statements below are executed (hint: the internal numeric code of character '`2`' is `50`).

(a) `System.out.println(a+a);`

[2]

(b) `System.out.println(a+b);`

[2]

(c) `System.out.println(b+c);`

[2]

(d) `System.out.println(c/d);`

[2]

(e) `System.out.println(c/(int)d);`

[2]

4. Angus wishes to write a program to change all occurrences of "organisation" within the text file `input.txt` to "organization", with all other characters unchanged. After the search and replace operation has been completed, the characters produced should be written to the file `output.txt`. Note that the given string "organisation" should be replaced irrespective of whether the string appears on its own as a separate word. This means that "organisational" should be changed to "organizational", for example.

Angus writes the following outline of a class:

```
import java.io.*;
import java.util.*;

public class SearchReplace
{  private static final String inputFile = "input.txt";
   private static final String outputFile = "output.txt";
   private static final String queryString = "organisation";
   private static final String replaceString = "organization";

   public static void main(String[] args)
   {  FileReader reader = null;
      PrintWriter out = null;
      try
      {  try
         {  ... // (A)
         }
         finally
         {  if (fr != null)
               fr.close();
            if (pw != null)
               pw.close();
         }
      }
      catch (IOException exception)
      {  System.out.println("I/O ERROR: " + exception);
      }
   }
}
```

(a) Write a segment of code that will open each of the files `input.txt` and `output.txt`, and read each line from `input.txt` in turn, storing the line in a `String` variable called `line`. No further processing on this variable is required at this stage. Assume that you code will be inserted into the above class at the position marked (A).

[5]

(b) Write a segment of code that will (i) replace every instance of `queryString` in a single line of `input.txt` (as stored in the `String` variable `line`) with `replaceString`, and (ii) output the modified line to `output.txt`. Indicate where your block of code should be placed in relation to your code from Part (a).

[10]

5. An administrator of a Javaball tournament wishes to write a Java program in order to output statistics relating to the teams taking part, after all matches in the tournament have been played.

Each Javaball match comprises two teams, and the objective is to win the match by scoring more goals than the opposition team. If the two teams score the same number of goals then the match is a draw.

If a team wins a Javaball match then it obtains 3 points; if a team loses then it obtains 0 points; whilst if a team draws then it obtains 1 point. Here are two example Javaball match results (the number of goals scored by each team is shown directly after the team name):

```
Abbey 3  Ford 2
Bournevale 5  Abbey 5
```

After these two matches, Abbey has 4 points, Bournevale has 1 point and Ford has 0 points. The following class is used to represent the details of a single team:

```
public class Team
{
   private String name;     // the team's name
   private int points;      // the team's points total

   public Team(String aName)
   {  name = aName;
      points = 0;
   }

   public String getName() // accessor method for name
   {  return name;
   }

   // accessor and mutator methods for points
   public int getPoints()
   {  return points;
   }

   public void setPoints(int numPoints)
   {  points = numPoints;
   }
}
```

The following class is used to represent a collection of teams that are involved in a given tournament.

```
public class TeamList
{
   // maximum possible number of teams
   private final int MAX_TEAMS = 100;

   // the list of teams in the tournament
   private Team [] teams;

   // actual number of teams in the tournament
   private int numTeams;

   public TeamList()
   {  teams = new Team[MAX_TEAMS];
      numTeams = 0;
   }

   // return a Team object with a given name
   public Team findTeam(String name)
   // (A)
   }

   // add a Team object with given name if not added already
   public void addTeam(String name)
   {  if (numTeams>=MAX_TEAMS)
         System.err.println("Too many teams");
      else
      {  Team team = findTeam(name);
         if (team==null)
         {  team = new Team(name);
            teams[numTeams] = team;
            numTeams++;
         }
      }
   }

   public int getNumTeams()  // accessor method for numTeams
   {  return numTeams;
   }

   public void printTeams()  // print out team information
    {  // (C)
   }
}
```

Assume that the main class containing the `main` method is called `JavaballMain`. Inside the `main` method the details of the Javaball tournament results are stored in a `String` object called `results`. Each Javaball result occupies a single line of `results`, and is formatted as follows:

```
<team><delimiter><goals><delimiter><team><delimiter><goals>
```

where `team` is a string comprising the name of a team, `delimiter` is a string, each character of which is a space, colon, comma or semi-colon, and `goals` is an integer. Assume that a team name consists of only one word that contains only upper or lower case members of the alphabet.

Here is an example `results` string:

```
Abbey 3, Ford 2
Bournevale: 5  ;  Abbey:  5
Park,  2 ;   Bracklinn, 1
```

Assume that `results` is in the correct format, and that there are at most 100 distinct teams in `results`.

(a) Complete the body of the method `findTeam` at point (A) above in the class `TeamList` (on Page 7). This method should search through the `teams` array looking for the first occurrence of a `Team` object whose name matches the `String` parameter `name`. If such an object is found it should be returned, otherwise `null` should be returned.

[6]

Suppose that the first few lines of main in `JavaballMain` are as follows:

```
String results = "Abbey 3, Ford 2\nBournevale: 5  ;  Abbey:
5\nPark,  2 ;   Bracklinn, 1"; // example only
TeamList teams = new TeamList();
String [] lines = results.split("\n");
int len = lines.length;
for (int i=0 ; i<len; i++)
{  String line = lines[i];
   // (B)
}
```

(b) Add code at point (B) above in `main` in order to perform the following tasks: (i) tokenize each line of `results` in order to obtain each Javaball match result, (ii) add each team name to the list of teams in `teamList` if necessary, and (iii) update the number of points for each team according to whether they have won, drawn or lost. Note that the particular value of `results` in the code segment above is only given as an example. Your code should not be reliant on this specific example and should be valid for any value of `results` (but you can assume that `results` is formatted correctly).

[12]

(c) Complete the body of the method `printTeams` at point (C) in the class `TeamList` (on Page 7) in order to print out each team name together with its current points total. The details for each team should occupy a single line. The team name should be left-justified using a field-width of 12 characters, and this should be followed by the points total, displayed using a field-width of 2 characters with leading zeros if required. The list of teams need not be sorted in any particular order. An example output from `printTeams` is as follows:

```
Abbey       04
Ford        00
Bournevale  01
Park        03
Bracklinn   00
```

[6]

(d) Add a method `getAverage()` to the class `TeamList`. This method should return the average number of points per team. With respect to the example following Question (c) above, the average number of points per team is 1.6. Give the header as well as the body of `getAverage`. If the number of teams is 0, then then 0 should be returned as the average.

[6]