Tutorial letter 103/1/2017

Interactive Programming

ICT2612

Semesters 1

School of Computing

IMPORTANT INFORMATION:

This tutorial letter contains important information
about your assignment 2.

DUE DATE: 18 APRIL 2017

UNIQUE CODE: 837390

INSTRUCTIONS

Work through all the questions and select the correct answer. When you are done, logon to myUNISA, select assignment 2 and complete and submit the online MCQ.

Question 1 **(1)**

Study the code below and indicate what the value of `total` will be.

```
int A = 15;

int B = 2;

float total = A/B;
```

(1) 0.0

(2) 7

(3) 7.0

(4) 7.5

Question 2 **(1)**

The data type _____ can store only a single Unicode character, e.g ‘A’ or ‘#’.

(1) `byte`

(2) `double`

(3) `char`

(4) `Boolean`

Question 3 **(1)**

Indicate which of the following declarations will render an **error**?

(1) `boolean result = true;`

(2) `int result = 10.0;`

(3) `int result = 'a';`

(4) `float result = 1.23;`

(5) `long result = 123_456_789;`

Question 4 **(1)**

Study the code below and indicate what the value of i_`total` will be.

```
double d_total = 123.5;

int i_total = (int)d_total;
```

(1) `123`

(2) `124`

(3) `123.0`

(4) `123.5`

(5) `error message`

Question 5 (1)

Which of the following is an **INVALID** variable name in Java?

(1) `String Key = "Java";`

(2) `String key = "Java";`

(3) `String 1key = "Java";`

(4) `String _key = "Java";`

(5) `String key$ = "Java";`

Question 6 (1)

Study the code below and indicate what the values of `val1` and `val2` will be:

```
int val1 = 5;
++val1;
int val2 = val1--;
```

(1) `val1: 5     val2: 4`

(2) `val1: 5   val2: 5`

(3) `val1: 5   val2: 6`

(4) `val1: 6   val2: 5`

(5) `val1: 6   val2: 6`

Question 7 (1)

Study the code below and indicate what the values of `answer` will be:

```
int x = 5, y = 6, z = 0, k = 2;

int answer = (x + y) * z / k;
```

(1) `0`

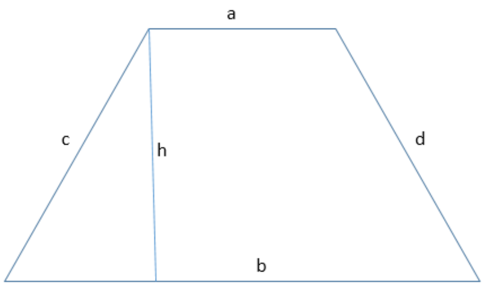
(2) `0.0`

(3) `5`

(4) `5.0`

(5) `Division by zero error message.`

Question 8 **(1)**



Below is the formula to calculate the area of the above trapezium (trapezoid):

Area = $= \frac{a+b}{2}$ x h

`The programmer created the following Java code to solve the above:`

```
double area;

double a=2, b=3, h=4;

area = calcArea(a,b,h);
```

Indicate which of the following Java code represents the correct code for the method `calcArea` that will solve the formula to calculate the area for a trapezium (trapezoid).

```
1. private static double calcArea(double a, double b, double h) {
   //calculate the area of a trapezoid
      double answer;
      answer = (a+b) / 2 x h;
     return (answer);
   }
```

```
2. private static double calcArea(double a, double b, double h) {
    //calculate the area of a trapezoid
      double answer;
      answer = (a+b)/2 * h;
     return (answer);
   }
```

```
3. private static double calcArea(double a, double b, double h) {
   //calculate the area of a trapezoid
      double answer;
      answer = (a+b) div 2 * h;
    return (answer);
   }
4. private static double calcArea(int a, int b, int h) {
    //calculate the area of a trapezoid
      double answer;
      answer = (a+b)/2 * h;
    return (answer);
   }
5. private static double calcArea(int a, int b, int h) {
    //calculate the area of a trapezoid
      double answer;
      answer = (a+b) div 2 * h;
    return (answer);
   }
```

Question 9 **(1)**

Study the code below and indicate what the values of `available`, `truth` and `canBorrow` will be.

```
String canBorrow = "No";

String isbn = "9-345-3445";

String status = "1";

boolean available = status == "1";

boolean truth = (isbn.substring(0,1).equals("9") && available);

if (truth) canBorrow = "Yes";
```

(1) `available: false`

`truth: false`

`canBorrow: No`

(2) `available: true`

`truth: false`

`canBorrow: No`

(3) `available: true`

`truth: true`

`canBorrow: No`

(4) `available: true`

`truth: true`

`canBorrow: Yes`

(5) `available: true`

`truth: false`

`canBorrow: Yes`

Question 10 **(1)**

Study the code below and indicate what the values will be for `hasMore`, `total` and `place`.

```
boolean hasMore = true;

int[] numbers = {1,2,4,3,2,1};

int place = numbers.length - 1, total = 0;


while (hasMore)

{

          total = total + numbers[place];

          place--;

          if (place == 0) hasMore = false;

}
```

(1) `total: 12 place: 0 hasMore: false`

(2) `total: 11 place: 0 hasMore: false`

(3) `total: 11 place: 0 hasMore: true`

(4) `total: 12 place: 1 hasMore: false`

(5) `total: 12 place: 1 hasMore: true`

Study the incomplete code below that is used to create the class Book.
Answer questions 11 to 16 that follow.

```
public class Car {

//instance variables

String brand, fueltype, manual_auto;

double price;
//constructors

public Car(String b, String f, String m, float p)
{
          //first constructor
          // (i)
          numCars++;
}

public Car(String brand, String fueltype, float price)
{
          //second constructor

          this.brand = brand;
          this.fueltype = fueltype;
          this.price = sell_price_special(price);
       numCars++;
}

private float sell_price(float p){
          return (float)(p * 1.10);
}//sell_price

private double sell_price_special(float p){
      //(ii)
}//sell_price_special

public String display(){
          String show = this.brand + "R " + this.price;
       return show;
}//display
}//Car
```

Question 11 (1)

Indicate which of the following will correctly link the variables for the first constructor **(i)** to the instance variables.

(1)

```
this.b = b;

this.f = f;

this.m = m;

this.p = sell_price(p);
```

(2)

```
this.b = brand;

this.f = fueltype;

this.m = manual_auto;

this.p = sell_price(p);
```

(3)

```
this.brand = b;

this.fueltype = f;

this.manual_auto = m;

this.price = sell_price(p);
```

(4)

```
String brand = b;

String fueltype = f;

String manual_auto = m;

float price = sell_price(p);
```

(5)

```
String b = b;

String f = f;

String m = m;

float p = sell_price(p);
```

Question 12 (1)

Indicate which of the following code **(ii)** will correctly calculate a discount of 10% and return the new discount price in the `sell_price_special()` method .

(1)

```
float discount, newprice;
  discount = (float) (p * 0.10);
  newprice = p / discount;
return newprice;
```

(2)

```
double discount, newprice;
  discount = (float) (p * 0.10);
  newprice = p - discount;
return  newprice;
```

(3)

```
double discount, newprice;
  discount = p / 1.10;
  newprice = p - discount;
return newprice;
```

(4)

```
double discount, newprice;
  discount = p / 1.10;
  newprice = discount;
return newprice;
```

(5)

```
double discount, newprice;
  discount = p / 1.10;
  newprice = discount;
return double(newprice);
```

Question 13 (1)

Indicate which of the following will correctly create two new instances (`car1` and `car2`) of the class `Car`.

(1)

```
Car car1 = new Car("mazda","petrol",150000);
Car car2 = new Car("nissan","diesel","manual",250000);
```

(2)

```
Car = car();
car1 = new Car("mazda","petrol",150000.00);
car2 = new Car("nissan","diesel","manual",250000.00);
```

(3)

```
car1 = new Car("mazda","petrol",150000);
car2 = new Car("nissan","diesel","manual",250000);
```

(4)

```
Car car1 = new Car(brand="mazda",fueltype="petrol",price=150000);
Car car2 = new Car(b="nissan",f="diesel",m="manual",p=250000);
```

(5)

```
Car car1 = new
           Car(brand="mazda",fueltype="petrol",price=150000.00);
Car car2 = new Car(b="nissan",f="diesel",m="manual",p=250000.00);
```

Question 14 **(1)**

Indicate what the value of `display` will be:

```
Car car3 = new Car("toyota","diesel",170000,"red");
String display = car3.display();
```

(1)

```
toyota R 153000.0
```

(2)

```
toyota R 187000.0
```

(3)

```
toyota R 153000.0 red
```

(4)

```
Error message.
The actual and formal argument list differ in length.
```

[✓ (5)]

```
Error message.
The constructor Car(String,String,int,String) is undefined.
```

Question 15 **(1)**

The purpose of the static variable `numCars` is to keep track of the number of instances of `Car`.

Indicate which of the following commands will correctly call this variable from the main program.

(1) Error message. You cannot call a static variable directly from the main program.

(2) `int numCars = Car.numCars;`

[✓ (3)] `int numCars = Car1.numCars;`

(4) `int numCars = Car1.display(numCars);`

(5) `int numCars = Car.display(numCars);`

Question 16 **(1)**

Indicate the value of `message`.

```
Car car1 = new Car("mazda","petrol",150000);

String message = car1.getFuelType();
```

(1) Error message. You cannot call the method getFuelType() from the main program.

(2) mazda R 135000.0 petrol

(3) mazda R 150000.0 petrol

(4) petrol mazda R 135000.0

[✓ (5)] petrol

Assume that the package `Arrays` is imported.

Study the code below and answer questions 15 to 20 that follow:

```
//declare arrays and allocate values to the arrays

String[] fueltypes = {"petrol","hybrid","electric","diesel"};

String[] transmission = {"manual","automatic"};

String[] drive = {"2WD","4WD","AWD"};

String[] brands = {"Mazda","Nissan","BMW","Toyota"};

char[] doors = {'2','3','4','5'};

String myDoors = new String(doors);
```

Question 17 **(1)**

Indicate which of the following statements can replace the array declaration that allocates the values to the array `brands`.

(1)

```
String[] brands = new String[4];
brands[0] = "Mazda";
brands[1] = "Nissan";
brands[2] = "BMW";
brands[3] = "Toyota";
```

(2)

```
String[4] brands;
brands[0] = "Mazda";
brands[1] = "Nissan";
brands[2] = "BMW";
brands[3] = "Toyota";
```

(3)

```
String[] brands[4];
brands[0] = "Mazda";
brands[1] = "Nissan";
brands[2] = "BMW";
brands[3] = "Toyota";
```

(4)

```
String[] brands[4];
brands[0] = new brands{"Mazda"};
brands[1] = new brands{"Nissan"};
brands[2] = new brands{"BMW"};
brands[3] = new brands{"Toyota"};
```

(5)

```
String[] brands;
brands[0] = new String{"Mazda"};
brands[1] = new String{"Nissan"};
brands[2] = new String{"BMW"};
brands[3] = new String{"Toyota"};
```

Question 18 (1)

Indicate which of the following code will correctly combine the array `brands` and the array `fueltypes` and initialise the array `cars`.

(1)
```
for (int i = 0; i < brands.length-1; i++){
  car[i] = brands[i] + " : " + fueltypes[i];
}
```

(2)
```
for (int i = 1; i < brands.length-1; i++){
  car[i] = brands[i] + " : " + fueltypes[i];
}
```

(3)
```
for (int i = 0; i < brands.length; i++){
  car[i] = brands[i] + " : " + fueltypes[i];
}
```

(4)
```
for (int i = 0; i < brands.length+1; i++){
  car[i] = brands[i] + " : " + fueltypes[i];
}
```

(5)
```
for (int i = 0; i < brands.length-1; i++){
  for (int j = i; j < brands.length; j++)
     car[i] = brands[i] + " : " + fueltypes[j];
}


  }
```

Question 19 (1)

Indicate the value of `myDoors`.

```
Error message:  incompatible types.
Reason:  you cannot create a new type String from type char[]
```

(1)
```
Error message:  incompatible types.

Reason:  you cannot create a new type String from type char[]
```

[✓ (2) `2345`]

(3) `2 3 4 5`

(4) `['2','3','4','5']`

(5) `[2345]`

Question 20 (1)

Indicate which of the following commands will correctly sort the array `drive` in ascending order.

(1) `Arrays.sort(drive,Collections Order());`

[✓ (2) `Arrays.sort(drive);`]

(3) `drive.sort();`

(4) `drive = drive.sort();`

(5) `Arrays(drive).sort();`

Question 21 (1)

Indicate which one of the following statements will correctly calculate the length of the array `drive`.

(1) `int len = drive.length();`

(2) `int len = drive.len();`

(3) `int len = length(drive);`

[✓ (4) `int len = drive.length;`]

(5) `int len = length.drive;`

Question 22 **(1)**

The programmer is requested to create code that will create a new user.

The new user must provide a user name and a password. The new password is entered twice and compared to ensure that they match.

```
String uName = requestUserInput();

String uPw1 = requestUserInput();

String uPw2 = requestUserInput();
```

Assume that the class `User` and the method `requestUserInput()` exist:

Class:

```
User(String username,

      String userPassword1)
```

Method:

`requestUserInput()`. This method requests the user to input information and returns a `String` to the calling program.

Indicate which one of the following options will **correctly** test for matching passwords and create a new instance of the User. Assume that the first and the second passwords are saved in String variables `uPw1` and `uPw2` respectively.

(1)
```
boolean isEqual = uPw1 == uPw2;


while not(isEqual)

{

  uPw1 = requestUserInput();

  uPw2 = requestUserInput();

  isEqual = uPw1 == uPw2;

}


//create new instance of user

User user1 = new User(uName, uPw1);
```

(2)

```
boolean isEqual = uPw1 == uPw2;

while <> isEqual
{
  uPw1 = requestUserInput();
  uPw2 = requestUserInput();
  isEqual = uPw1 == uPw2;
}

//create new instance of user
User user1 = new User(uName, uPw1);
```

(3)

```
while <> (isEqual)
{
  uPw1 = requestUserInput();
  uPw2 = requestUserInput();
  isEqual = isEqual = uPw1 == uPw2;
}

//create new instance of user
User user1 = new User(uName, uPw1);
```

(4)

```
boolean isEqual = uPw1.equals(uPw2);

while <> (isEqual)
{
  uPw1 = requestUserInput();
  uPw2 = requestUserInput();
  isEqual = isEqual = uPw1.equals(uPw2);
}

//create new instance of user
```

```
User user1 = new User(uName, uPw1);
```

(5)

```
boolean isEqual = uPw1.equals(uPw2);

while (!isEqual)
{
   uPw1 = requestUserInput();
   uPw2 = requestUserInput();
   isEqual = uPw1.equals(uPw2);
}

//create new instance of user
User user1 = new User(uName, uPw1);
```

Question 23 (1)

Study the code and indicate the value of `place`.

```
String id = "640423";
int place = id.indexOf("4");
```

(1) 1

(2) 2

(3) 1 3

(4) 2 4

(5) 24

Question 24 (1)

Which one of the following is **NOT** an example of an *exception* error that can occur in Java?

(1) A user entered invalid data.

(2) A file that needs to be opened cannot be found.

(3) A network connection has been lost in the middle of communications.

(4) `The JVM has run out of memory.`

[✓ (5) `The programmer entered the incorrect code to create a button.`]

Question 25 (1)

Study the incomplete code and indicate which of the following code **(i)** in the method `detStatus()` will return the string “pass” or “fail” depending on the marks.

```
double marks = 34;
String status = detStatus(marks);
. . . .

private static String detStatus(double marks) {
//method to determine the status (pass or fail)

  String status;
  //(i)

return status;
}//detStatus
```

(1) `if (marks < 50) status = "fail";  status = "pass";`

(2) `if (marks < 50) status = "fail" else status = "pass";`

(3)
```
switch(marks){
  case < 50: status = "fail"; break;
  case >= 50: status = "pass"; break;
}
```

[✓ (4) `status = marks >= 50 ? "pass" : "fail";`]

(5) `status (marks >= 50)? {"pass"}{ "fail"}`

Question 26 (1)

When entering the code below, the programmer receives an error message.

```
int[] daysInMonth = {31,28,31,30,31,30};
int totalDays = 0;
```

```
for (int i=0; i < 12; i++){
        totalDays = totalDays + daysInMonth[i];
}
```

Indicate which one of the following options will intercept the error without crashing the program.

```
(1) for (int i=0; i < 12; i++){
      try{
            totalDays = totalDays + daysInMonth[i];
            } catch (Exception e){boolean error = true;}
    }
```

```
(2) for (int i=0; i < 12; i++){
      try{
            totalDays = totalDays + daysInMonth[i];
            } (catch e){boolean error = true;}
    }
```

```
(3) for (int i=0; i < 12; i++){
      try{
            totalDays = totalDays + daysInMonth[i];
            } exception (Catch e){boolean error = true;}
    }
```

```
(4) for (int i=0; i < 12; i++){
      try{
            totalDays = totalDays + daysInMonth[i];
            } catch (Exception e)
    }
```

```
(5) for (int i=1; i < 12; i++){
      try{
            totalDays = totalDays + daysInMonth[i];
```

```
         } catch e {boolean error = true;}
}
```

Study the code below and answer question 27 to 29.

Assume that the method `get_user_input()` exists. This method requests the user to enter his/her blood type and returns the `bloodtype` as a String to the calling program.

```
String bloodType = get_user_input();
String message = null;
switch (bloodType){
  case "A" : message = "You can donate to A and B"; break;
  case "B" : message = "You an donate to B and AB";
  case "AB" : message = "You can donate to other AB's";
  case "O" : message = "You can donate to any bloodtype";
}

String output = bloodType + ": " + message;
```

Question 27 **(1)**

What will be the value of `output` be if `bloodType` is initialised to `"A"`.

[✓ (1)] `B: You can donate to A and B`

(2) `B: You can donate to B and AB`

(3) `B: You can donate to other AB`

(4) `B: You can donate to any bloodtype`

(5) Error message. There is not a '`break`' clause after the case statement that tests for `"B"`.

Question 28 **(1)**

What will be the value of `output` be if `bloodType` is initialised to "`AB`"?

(1) `B: You can donate to A and B`

(2) `B: You can donate to B and AB`

(3) `B: You can donate to other AB`

(4) `B: You can donate to any bloodtype`

(5) `Error message. A case statement can only test for a single character such as A, B or O, but not for AB.`

Study the code below to answer the question that follows.

```
int day = 1;

String dayName;

switch(day){

  case 1: dayName = "Sun"; break;

  case 2: dayName = "Mon"; break;

  case 3: dayName = "Tue"; break;

  case 4: dayName = "Wed"; break;

  case 5: dayName = "Thu"; break;

  case 6: dayName = "Fri"; break;

  default: dayName = "Sat";

}
```

Question 29 **(1)**

Which one of the following options can replace the above `switch` statement and still render the same results?

(1)

```
int day = 1;

String dayName;

String daysInWeek[7]  =

                {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};

try{

         dayName = daysInWeek[day];

}

catch (Exception e){dayName = "Sat";}
```

(2)

```
int day = 1;

String dayName;
```

```
    String daysInWeek[7]  =
                    {"","Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    try{
              dayName = daysInWeek[day];
    }
    catch (Exception e){dayName = "Sat";}
```

```
(3) int day = 1;
    String dayName;
    String[] daysInWeek  =
                    {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    try{
              dayName = daysInWeek[day];
    }
    catch (Exception e){dayName = "Sat";}
```

```
(4) int day = 1;
    String dayName;
    String[] daysInWeek  =
                {"","Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    try{
              dayName = daysInWeek[day];
    }
    catch (Exception e){dayName = "Sat";}
```

```
(5) int day = 1;
    String dayName;
    String[8] daysInWeek  =
                    {"","Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    try{
              dayName = daysInWeek[day];
    }
    catch (Exception e){dayName = "Sat";}
```

Question 30 (1)

Which one of the following is NOT an access modifier in Java?

(1) `private`

(2) `protected`

(3) `public`

(4) `void`

Question 31 (1)

Consider the three lines of code below.

```
String s = "30564.5";           //line 1
char[] cArray = s.toCharArray(); //line 2
int answer = cArray.length;      //line 3
```

Which one of the following options is correct regarding the above three statements?

(1) No compiler error and the value of `answer` is 7.

(2) Compiler errors in line 1 and line 2.

(3) No compiler error and the value of `answer` is 6.

(4) Compiler error in line 1.

(5) Syntax error in line 2.

Question 32 (1)

Study the code below and indicate the result of `f_answer`.

```
double d_answer = 123.4;
float f_answer = (int)d_answer;
```

(1) `123`

(2) `123.0`

(3) `124`

(4) `124.0`

(5) Syntax error. Cannot cast an integer to a float value.

Question 33 (1)

Indicate the value of result:

```
int val1 = 11;

int val2 = 2;

double result = val1 % val2;
```

(1) Error message.

Reason: There is no such operator as %.

(2) Error message.

Reason: you cannot assign the result of % to a double value.

[✓ (3) 1.0]

Reason: the remainder of 11 divided by 2 is 1.

(4) 5

Reason: 11 divided by 2 is 5.

(5) 5.0

Reason: 11 divided by 2 is 5.0, displayed as type double.

Question 34 (1)

Study the code below and indicate the value of place.

```
String email = "john.nell@gmail.com";

int place = email.indexOf("@");
```

(1) Error message. You should search for '@'

(2) 8

[✓ (3) 9]

(4) 10

(5) @

Question 35 **(1)**

Study the code below and answer the question that follows.

```
public class Compare {

          static int larger(int x, int y){
             if (x > y) return x;
             else return y;
          }

          static char larger(char x, char y){
             if (x > y) return x;
             else return y;
          }

          static double larger(double x, double y){
             if (x > y) return x;
             else return y;
          }

          static String larger(String x, String y){
             if (x.length() > y.length()) return x;
             return y;
          }
}
public class Main {
          public static void main(String[] args) {

          Compare c = new Compare();
          int s1 = c.larger(10,20);
          char s2 = c.larger('a', 'b');
```

```
            String s3 = c.larger("Hope", "hope");
            double s4 = c.larger(10.1, 11);
            }
}
```

Which one of the following is correct regarding the above code?

(1) Java does not allow multiple methods with the same name in a class.

(2) The values for s1,s2,s3 and s4 are:

```
s1: 20
s2: b
s3: hope
s4: 11.0
```

(3) The values for s1,s2,s3 and s4 are:

```
s1: 20
s2: b
s3: Hope
s4: 11.0
```

(4) The values for s1,s2,s3 and s4 are:

```
s1: 20
s2: b
s3: Hope
s4: 11
```

(5) The statement

```
double s4 = c.larger(10.1, 11);
```

gives a compiler error because there is no function that matches the

`arguments (double, int).`

Question 36 **(1)**

Below is an incomplete Java program.

```
import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;


public class Main {


         public static void main(String[] args) throws IOException {

         //(i)


         }

}
```

You are given the task of counting how many employees have `gmail` email addresses. There is a file named "`info.txt`".

Each employee's information is captured in a single line, e.g.

```
Smith, JH. 6202125080082. HR. john.smith123@gmail.com
```

You can assume that only one email address is included in a line of text and all the required packages are imported.

Which one of the following achieves this task correctly **`(i)`** in Java and will have the correct value in `count`?

(1)
```
BufferedReader br = new BufferedReader(fr);

int count = 0;

String line = br.readLine();

while !(line=null){
  if (line.contains("@gmail.com")) count++;
  line = br.readLine();
}
br.close();
```

(2)

```
FileReader fr = new FileReader("info.txt");
BufferedReader br = new BufferedReader(fr);
int count = 0;
String line = br.readLine();
while (line!=null){
  if (line.contains("@gmail.com")) count++;
  line = br.readLine();
}
br.close();
fr.close();
```

(3)

```
FileReader fr = new FileReader("info.txt");
int count = 0;
String line = fr.readLine();
while (line!=null){
  if (line.contains("@gmail.com")) count++;
  line = fr.readLine();
}
fr.close();
```

(4)

```
FileReader fr = new FileReader("info.txt");
BufferedReader br = new BufferedReader(fr);
int count = 0;
String line = br.readLine();
while (line!=null){
  if (line.contentEquals("@gmail.com")) count++;
  line = br.readLine();
}
br.close();
```

```
    fr.close();
```

(5)

```
    FileReader fr = new FileReader("info.txt");
    BufferedReader br = new BufferedReader(fr);
    int count = 0;
    String line = br.readLine();
    while NOT(line = null){
      if (line.contentEquals("@gmail.com")) count++;
      line = br.readLine();
    }
    br.close();
    fr.close();
```

Question 37 **(1)**

Study the incomplete code below used by a rental car agency.

```
boolean license = getLicense();
int age = getAge();
String MF = getMF();
boolean fines = getFines();

boolean valid = false;
// (i)
```

In the agency the rules are enforced when a person apply to rent a car:

- The person must have a valid driver's license
- A female applicant must be 23 years and older
- A male applicant must be 25 years and older
- The person may not have any outstanding traffic fines

Which one of the following expression will **NOT** enforce the rules above?

Assume that all the variables used in the options are declared and initialised correctly.

(1)

```
valid = ((license==true) && (fines==false) &&
        (((age>=23) && (MF=="F" ))||( (age>=25) && (MF == "M"))));
```

(2)

```
valid = ((license) && !(fines) &&
        (((age>=23) && (MF=="F" ))||( (age>=25) && (MF == "M"))));
```

(3)

```
valid = ((license==true) && (fines==false) &&
         (((age>=23) && (MF.equals("F") ))||
             ( (age>=25) && (MF.equals("M")))));
```

(4)

```
valid = ((license) && !(fines) &&
         (((age>=23) && (MF.equals("F") ))||
             ( (age>=25) && (MF.equals("M")))));
```

(5)

```
valid = ((license=true) && (fines=false) &&
        (((age>=23) && (MF="F" ))||( (age>=25) && (MF = "M"))));
```

Question 38 **(1)**

Choose the correct data types for the variables in the code below:

```
height = 165;
weight = 70.3;
bmi = weight / height / height * 10000;
```

(1)

```
float height = 165;
float weight = 70.3;
float bmi = weight / height / height * 10000;
```

(2)

```
double height = 165;
double weight = 70.3;
float bmi = weight / height / height * 10000;
```

(3)

```
float height = 165;

double weight = 70.3;

float bmi = weight / height / height * 10000;
```

(4)

```
float height = 165;

float weight = 70.3;

dubble bmi = weight / height / height * 10000;
```

(5)

```
float height = 165;

double weight = 70.3;

double bmi = weight / height / height * 10000;
```

Question 39 **(1)**

Which one of the following statements explains the contents of `activity_main.xml` file in an Android application project?

(1) It contains information about the sound and image files used in the application.

(2) It contains information about the default Graphical User Interface of the application.

(3) It contains the code of the class `MainActivity`.

(4) It contains a summary of all the setup information of the application.

(5) It contains a summary of all the XML files used in the application.

Question 40 **(1)**

A _____variable, that is not accessible by any other method, class or program is defined inside a method, constructor or block and will be destroyed when the method has completed.

(1) local

(2) instance

(3) class

(4) open

(5) close

Question 41 **(1)**

The ______ of a class are instructions that the class uses to manipulate values, generate output or perform various actions.

(1) attributes

(2) classes

(3) overloading

[✓ (4) methods]

(5) objects

Question 42 **(1)**

In object-oriented terminology, the characteristics of an object are defined by its ____

[✓ (1) attributes]

(2) classes

(3) instances

(4) triggers

(5) variables

Question 43 **(1)**

Which is the correct keyword that indicates inheritance between two classes in Java?

(1) attributes

(2) enlarge

[✓ (3) extends]

(4) inherits

(5) super

Question 44 **(1)**

Which one of the following statements describes **inheritance** correctly?

(1) It allows generic code to be placed in a superclass and more specialized code in subclasses, thus promoting code reuse.

(2) It is used when a subclass is more general than a superclass thereby creating a class hierarchy.

(3) The primary reason for using it is to reduce execution times of programs.

(4) Using it reduces errors because you can simply copy and paste code from a superclass to new subclasses.

(5) The primary reason for using it is to make the code more “human” readable.

Question 45 **(1)**

___ is the capability of an object to have data and functionality available to the user, without the user having to understand the implementation within the object.

(1) Aggregation

(2) Encapsulation

(3) Inheritance

(4) Polymorphism

(5) Visibility

Question 46 **(1)**

Indicate which one of the following best describes a **public access modifier**:

(1) Only allows access from inside the same class.

(2) Allows access inside the class, subclass or other classes of the same package as the modifier.

(3) Allows access from inside the same package.

(4) Allows access from anywhere, inside and from outside the package.

(5) None of the above.

Question 47 **(1)**

In an Android project the _____ folder includes the resource files (images, music and videos) for the project.

(1) `assets`

(2) bin

(3) gen

[✓ (4) res]

(5) src

Question 48 **(1)**

In an Android project the _____ folder includes the code source files for the project

(1) assets

(2) bin

(3) gen

(4) res

[✓ (5) src]

Question 49 **(1)**

The ____ package of Java provides classes for performing arithmetic operations.

(1) java.io

(2) java.calc

(3) java.lang

[✓ (4) java.math]

(5) java.util

Question 50 **(1)**

The data type Boolean can store data upto ___ that store(s) true / false flags.

[✓ (1) 1 bit]

(2) 2 bits

(3) 1 byte (8 bits)

(4) 2 bytes

(5) 4 bytes

TOTAL: 50

2017