UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2016

IS52028A
Principles and Application of of Programming

Duration: 2 hours 15 minutes

Date and time:

This paper is in three parts: part A (Multiple Choice), part B (Java) and part C (C++). You should answer one question from part A (either Java or C++) and TWO questions from either part B OR part C. Part A carries 40 marks, and each question from parts B and C carries 30 marks. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 100 marks available on this paper.

**THIS PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM**

Part A

Multiple choice

Question 1 JAVA: Each question has one correct answer

(a) Which of these keywords is used to refer to member of base class from a sub class?

i. upper
ii. super
iii. this
iv. None of the mentioned

[4]

(b) Which of these keywords can be used to prevent Method overriding?

i. static
ii. constant
iii. protected
iv. final

[4]

(c) Which of the following statements are incorrect?

i. public members of a class can be accessed by any code in the program.
ii. private members of a class can only be accessed by other members of the class.
iii. private members of a class can be inherited by a sub class, and become protected members in sub class.
iv. protected members of a class can be inherited by a sub class, and become private members of the sub class.

[4]

(d) Which of these is supported by method overriding in Java?

i. Abstraction
ii. Encapsulation
iii. Polymorphism
iv. None of the mentioned

[4]

(e) What is the process of defining two or more methods within same class that have same name but different parameter declarations?

i. method overloading
ii. method overriding

iii. method hiding
iv. None of the mentioned

[4]

(f) Which of these is not a correct statement?

i. Every class containing abstract methods must be declared abstract.
ii. Abstract classes can contain abstract and concrete methods
iii. Abstract class can be inherited.
iv. Abstract class can be instantiated by new operator.

[4]

(g) The concept of multiple inheritance is implemented in Java by

i. extending two or more classes
ii. extending one class and implementing one or more interfaces
iii. implementing one interface or extending one or more classes
iv. all of these

[4]

(h) The fields in an interface are implicitly specified as,

i. static only
ii. protected
iii. static and final
iv. private

[4]

(i) The following program has a compilation error. True or False?

```
class Superclass
{
    void foo()
    {
        System.out.println("Superclass.foo");
    }
}

class Subclass extends Superclass
{
    private void foo()
    {
        System.out.println("Subclass.foo");
    }
}
```

[4]

(j) The following program has no compilation error. True or False?

```
public class Test {
  public static void main(String[] args) {
    A a = new A();
    a.print();
  }
}

class A {
  String s;

  A(String newS) {
    s = newS;
  }

  void print() {
    System.out.println(s);
  }
}
```

[4]

Question 2 C++: Each question has one correct answer

(a) C++ is...

i. compiled
ii. interpreted
iii. just in time compiled
iv. None of the mentioned

[4]

(b) Which of these keywords is used to refer to member of base class from a sub class?

i. upper
ii. super
iii. this
iv. None of the mentioned

[4]

(c) How is the parameter a in the following function passed?

```
void f(int & a){

}
```

i. by value
ii. by reference
iii. by const reference
iv. by pointer

[4]

(d) Which of the following correctly iterates through all elements of the `std::vector<std::string> v`?

i.
```
for(auto i : v){
    std::cout << i << std::endl;
}
```

ii.
```
for(int i : v){
    std::cout << i << std::endl;
}
```

iii.
```
for(int i = 0; i < v.length; i++){
        std::cout << v[i] << std::endl;
    }
```

iv.
```
    for(auto i = 0; i < v.length; i++){
        std::cout << v[i] << std::endl;
    }
```

[4]

(e) What is the type of variable a?

```
vector <int> v;
auto a = v.begin();
```

i. `auto`
ii. `int`
iii. an iterator type
iv. a `unique_ptr` type

[4]

(f) What is the error in this code?

```
int a, b;
std::cin >> a >> b;
```

i. b should be declared a std::string
ii. you cannot have two `>>` operators on a single line
iii. it should be `<<` instead of `>>`
iv. nothing, the code is correct

[4]

(g) Which constructors is called when this variable (of class MyClass) is created?

```
MyClass c;
```

i. no constructor
ii. a constructor taking an int parameter
iii. a default constructor
iv. a copy constructor

[4]

(h) What do we declare with `float* p, p1;`

i. A float pointer p and a float pointer p1.
ii. A float pointer p and a float p1.
iii. A float p and a float pointer p1.
iv. A float p and a float p1.

[4]

(i) Pure substitution in C++ inheritance means:

i. Inheritance overriding only base-class functions
ii. You can add more functionality in the derived class
iii. Private inheritance
iv. None of the above

[4]

(j) An implementation of an assignment operator should:

i. Check for self-assignment
ii. Perform memory-management
iii. All of the above
iv. None of the above

[4]

Part B

Question 3 Inheritance & Polymorphism

(a) Briefly explain the concepts of (i) inheritance and (ii) polymorphism in Java with a sentence or two. [4]

(b) You are given the following code, that you can assume compiles without errors.

```
class SuperClass
{
     protected int x = 0;
     public SuperClass(int x) {
         this.x = x;
     }
     private void increment() {
          x++;
     }
     protected final void add(int y) {
          x += y;
     }
     public void display(){
           System.out.println(x);
     }
}
public class SubClass extends SuperClass
{
     public SubClass(int x){
         super(x);
      }
     public void display(){
           add(2);
           super.display();
     }
     public static void main(String [] args){
           SuperClass sc = new SuperClass(3);
           sc.display();
           sc = new SubClass(3);
           sc.display();
     }
}
```

i. List the name of all methods Subclass inherit from SuperClass.

ii. List the name of all methods of SuperClass that are visible in Subclass (in other words, methods that can be called directly).

iii. List the name of all methods of SuperClass that may NOT be overridden by Subclass.

iv. What gets displayed on the screen when SubClass is executed?

[8]

(c) You are given the following code and inheritance / composition relationships.

- a Motorized Vehicle (`MotorVeh`) is-a Vehicle (`Vehicle`)
- a Bicycle (`Bicycle`) is-a Vehicle (`Vehicle`)
- a Car (`Car`) is-a Motorized Vehicle (`MotorVeh`)
- a Truck (`Truck`) is-a Motorized Vehicle (`MotorVeh`)
- a Motorized Vehicle (`MotorVeh`) has-an Engine (`Engine`)

(i) Write a Java code that captures this relationships. [5]

(ii) Extend the code above as follows:

- An `Engine` has a member `double Speed` which indicates the speed that this engine achieves. This should be set during initialization of an `Engine` object. The `Engine` also returns the Engine speed via a member function `double getSpeed()`.
- A `MotorVeh` should initialize its `Engine` in the constructor (along with the Engine's speed). A member function `float getEngineSpeed()` should return the speed of the engine.
- A `Bicycle` is not a Motorized Vehicle and therefore does not have an engine. It does have a `Speed` member though that is initialized during the construction of the object and has a default value of 5.0.
- A `Truck` and a `Car` should make use of the Motorized Vehicle constructor in order to initialize their engine. The constructor of each of these methods should take an argument `Speed` to do so, with a default value of 20.0;
- A `Truck`, `Car` and `Bicycle` should implement the function `double getVehSpeed()`, that returns the speed of the vehicle. This function should be handled in a polymophic manner, and should also make the `Vehicle` class an abstract base class. Note that a `Bicycle` can achieve 100% of its speed, while a `Truck` only 70% and a `Car` only 90%.

[9]

(iii) Write a function `double getVehSpeed(Vehicle v) {...}` that returns the speed of a vehicle, be it a Bicycle, Car or Truck. Show how you would utilize this function with a `Truck`, `Car` and `Bicycle` object. [4]

Question 4 Data Structure

(a) What output does the following program produce? You can assume that the program compiles and executes without producing any errors.

```
public class Myclass {

 private void method(ArrayList<String> s) {
    Stack<String> words = new Stack<String>();
    Iterator<String> iter = s.iterator();
    while (iter.hasNext()){
       words.push(iter.next());
    }
    String result = "";
    while (!words.isEmpty()) {
        result = result + " " + words.pop();
    }
    System.out.println(result);
  }
 public static void main (String[] args) {
    ArrayList<String> s = new ArrayList<String>();
    s.add("here");
    s.add("comes");
    s.add("summer");
    Myclass m = new Myclass();
    m.method(s);
  }
}
```

[7]

(b) Given the following program:

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

  public class Words {
    static public void main(String[] args) {
    Set<String> words = new HashSet<String>();
    String delim = " \t\n.,:;?!-/()[]\"\’";
    String line;
    int count = 0;
    try{
        FileReader fin = new FileReader("test.txt");
        Scanner in = new Scanner(fin);

        while (in.hasNextLine()) {
          line = in.nextLine();
          StringTokenizer st =
             new StringTokenizer(line, delim);
          while (st.hasMoreTokens()) {
            count++;
            words.add(st.nextToken().toLowerCase());
          }
        }
    } catch (Exception e) {System.out.println(e.toString());}
        System.out.println(count);
       System.out.println(words.size());
  }
}
```

What does this program do and what should be the output of the program in (a) if the ’test.txt’ contains the following:

```
Studying studying at Goldsmiths.
Working and studying at  goldsmiths.
```

[8]

(c) Write a complete program that stores the words in the file "test.txt" and the number of their occurences in a HashMap. Use the iterator through the HashMap to print each word and the number of its occurrences. The output of the program should be:

```
at 2
working 1
goldsmiths 2
studying 3
and 1
```

[10]

(d) What would the output of the program in (c) be if TreeMap was used instead of HashMap?

[5]

Question 5 Recursion and DataBase

(a) Consider the following method:

```
/* @param x >=0*/
public void mystery(int x)
{
  System.out.print(x % 10);
  if ((x / 10) != 0)
  {
      mystery(x / 10);
  }

}
```

What is the output of of mystery(1234)? [5]

(b) Given the following program 'Q3.java':

```
import java.sql.*;
public class  Q3 {
   public static void main(String args[]) {
     String
              driver = "com.mysql.jdbc.Driver",
              url = "jdbc:mysql://igor.gold.ac.uk",
              database = "mas01lo_java",
              user = "mas01lo",
              password = "xxxxxxx",
              table = "test_table";
     try
       {  Class.forName(driver);
          Connection db = DriverManager.getConnection(url + "/" + \\
                                          database, user, password);
        try
          {
          db.createStatement().executeUpdate("CREATE TABLE " + table +\\
                           " (id int PRIMARY KEY, name text, dateOfBirth date)");
          db.createStatement().executeUpdate("INSERT INTO " + table +\\
               "(id, name, dateOfBirth VALUES (1, 'John Smith', '1995-12-06')");
          db.createStatement().executeUpdate("INSERT INTO " + table + \\
             "(id, name, dateOfBirth) VALUES (2, 'Lahcen Ouarbya', '1976-11-12')"
          ResultSet rs = db.createStatement().executeQuery("SELECT * FROM "+table
           while (rs.next()) System.out.println(rs.getString(1));

           ResultSetMetaData rsmd = rs.getMetaData();
           int numberOfColumns=rsmd.getColumnCount();
           rs.close();

           //---------------------- Add your code for part c here -------------

           db.createStatement().executeUpdate("DROP TABLE " + table + "");
           }
           catch (SQLException e) { System.out.println("Error 1"); }
           db.close();
         }
       catch (ClassNotFoundException e) {System.out.println("Error 2");}
       catch (SQLException e) { System.err.println("Error 3");}
   }
}
```

i. What is the output of this program?

ii. What would the output be if the connection to the database is failed?

iii. What would the output be if the driver was not found?

iv. What would the output be if there was an error in the SQL query?

[15]

(c) Write down the missing code fragment in program so that the output is the following:

```
id,    name,           dateOfbirth
1,    John Smith,       1995-12-06
2,   Lahcen Ouarbya,    1976-11-12
```

[10]

Part C

Question 6 Pointers

(a) Give two advantages and two disadvantages of declaring a variable as each of the following [12]

  i. value
  ii. pointer
  iii. reference

(b) Explain the difference between stack and heap memory allocation [8]

(c) Look at the following code and describe how and when memory is allocated and deallocated for each of the variables and the heap memory associated with each variable. In your answer refer to heap and stack memory. [10]

```
#include <iostream>


int *makePointer(int size)
{
    int *p = new int[size];
    for (int i = 0; i < size; i++){
        p[i] = i;
    }
    return p;
}


int main(int argc, const char * argv[]) {
    int s = 5;
    int * a;
    a = makePointer(s);

    std::cout << a[2];
    delete a;

    return 0;
}
```

Question 7 std::vector & concurrency

(a) Describe the `std::vector` and how it is used. [5]

(b) Give three advantages of using a `std::vector<int>` rather than an array (`int []`) [3]

(c) Write a complete program that reads in the words in the file "test.txt" and counts the number of their occurences. it shold then print each word and the number of its occurrences.

If the 'test.txt' contains the following:

```
studying studying at goldsmiths.
working and studying at  goldsmiths.
```

The output of the program should be:

```
  studying 3
  at 2
  goldsmiths. 2
  working 1
  and 1
```

[14]

(d) You are given the following code snippet, which compiles and runs without errors. The code implements a character type that can transfer energy to other character objects.

```
class character
{
     std::mutex m;
     float energy;
public:
      void transfer(character& fromC, character& toC, float amount)
     {
          std::lock_guard<std::mutex> lock_from(fromC.m);
          std::lock_guard<std::mutex> lock_to(toC.m);
          fromC.energy -= amount;
          toC.energy += amount;
          cout << fromC.energy << endl;
          cout << toC.energy << endl;
     }
};
```

(i) Can you see any potential risks in using this code? [4]

(ii) Modify the code above so that it runs without the risk of deadlocks or race conditions, elaborating on your solution. [4]

Question 8 Inheritance & Polymorphism

(a) Briefly explain the concepts of (i) inheritance and (ii) polymorphism in C++ with a sentence or two. [4]

(b) You are given the following code, that you can assume compiles without errors.

```
#include <iostream>
using namespace std;

class A
{
  int* a;
  public:
     A() {a = new int(3);}
     ~A() { cout << " A " ;  delete a; a = nullptr; }};

class B : public A
{
     int* b;
  public:
     B() {b = new int(3);}
     ~B() { cout << " B "; delete b; b = nullptr; }};

class C : public B {
     int* c;
  public:
     C() {c = new int(3);}
     ~C() { cout << " C "; delete c; c = nullptr; }};

int main() {
     A* ptr = new C();
     delete ptr;
     ptr = nullptr;}
```

(i) What is the output printed to screen? [2]

(ii) Does the code cause any memory leaks? If so, how would you fix them? [4]

(iii) Sort the classes A, B and C based on the size in bytes, in decreasing order (largest size first), elaborating on your choice. [2]

(c) You are given the following code and inheritance / composition relationships.

- a Motorized Vehicle (`MotorVeh`) is-a Vehicle (`Vehicle`)
- a Bicycle (`Bicycle`) is-a Vehicle (`Vehicle`)
- a Car (`Car`) is-a Motorized Vehicle (`MotorVeh`)
- a Truck (`Truck`) is-a Motorized Vehicle (`MotorVeh`)
- a Motorized Vehicle (`MotorVeh`) has-an Engine (`Engine`)

(i) Write the code in C++ that captures the following relationships. [5]

(ii) Extend the code above as follows: [9]

- An `Engine` has a member `float Speed` which indicates the speed that this engine achieves. This should be set during initialization of an `Engine` object. The `Engine` also returns the Engine speed via a member function `float getSpeed()`.
- A `MotorVeh` should initialize its `Engine` (Hint: use a pointer to store Engine) in the constructor (along with the Engine's speed). A member function `float getEngineSpeed()` should return the speed of the engine.
- A `Bicycle` is not a Motorized Vehicle and therefore does not have an engine. It does have a `Speed` member though that is initialized during the construction of the object and has a default value of 5.0.
- A `Truck` and a `Car` should make use of the Motorized Vehicle constructor in order to initialize their engine. The constructor of each of these methods should take an argument `Speed` to do so, with a default value of 20.0;
- A `Truck`, `Car` and `Bicycle` should implement the function `float getVehSpeed()`, that returns the speed of the vehicle. This function should be handled in a polymophic manner, and should also make the `Vehicle` class an abstract base class. Note that a `Bicycle` can achieve 100% of its speed, while a `Truck` only 70% and a `Car` only 90%.

(iii) Write a function `float getVehSpeed(Vehicle* v)` {...} that returns the speed of a vehicle, be it a Bicycle, Car or Truck. Show how you would utilize this function with a `Truck`, `Car` and `Bicycle` object. [4]