UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2017

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 either TWO questions from part B (Java) or TWO questions from part C (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) What is garbage collection in the context of Java?

i. the operating system periodically deletes all of the java files available on the system

ii. any package imported in a program and not used is automatically deleted

iii. when all references to an object are gone, the memory used by the object is automatically reclaimed

iv. the JVM checks the output of any Java program and deletes anything that doesn't make sense

[4]

(b) Which collection class associates values with keys, and orders the keys according to their natural order?

i. java.util.HashSet

ii. java.util.LinkedList

iii. java.util.TreeMap

iv. java.util.SortedSet

[4]

(c) Methods that have the same name, but different Parameter lists and different definitions are known as

i. overriding

ii. constructor

iii. overloading

iv. none of the above

[4]

(d) If b is a variable of the class Ball (below), how would you call the function draw? [4]

```
class Ball{
      private  int radius;

       public Ball(int r){
          radius = r;
         }

      public void draw(int x, int y);
}

Ball b = new Ball(6);
```

i. `b.draw(10, 20);`

ii. `b->draw(10, 20);`

iii. `Ball.draw(10, 20);`

iv. `Ball->draw(10, 20);`

(e) Interfaces help in which type of inheritance.

i. multiple inheritance

ii. multilevel inheritance

iii. hierarchical inheritance

iv. none of the above

[4]

(f) All methods of interface are public and abstract.

i. true

ii. false

[4]

(g) What are the functions of the dot(.) operator? More than one choice might apply

i. It enables you to access instance variables of any objects within a class

ii. It enables you to store values in instance variables of an object

iii. It is used to call object methods

iv. It is used to create a new object

[4]

(h) Which of the following declares an array of string objects?

i. String[ ] s;
ii. String s [ ] ;
iii. String [s] ;
iv. none of the above

[4]

(i) A class member declared protected becomes a member of a subclass of which type?

i. public member
ii. private member
iii. protected member
iv. none of the above

[4]

(j) Assuming the class A and class B given below are defined in different packages, packageA and packageB.

```
package packA
public class A
{
    protected void msg(){
         System.out.println("Hello");
    }
}

package packB;
import packA;
class B extendsA
{
     public static void main(String [] args){
          B obj=newB();
          obj.msg();
     }
}
```

Which of the following is correct about the class B?

i. it has a compilation error

ii. it has a run time error

iii. it compiles, runs and prints the string ‘Hello’

iv. none of the above

[4]

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

(a) What is a pointer? [4]

i. a template

ii. a memory address

iii. an object that is stored on the heap

iv. a float

(b) What is the type of variable a? [4]

```
vector <int> v = {1, 2, 3, 4};
auto a = &v[0];
```

i. `auto`

ii. `int`

iii. `int &`

iv. `int *`

(c) What does this code do? [4]

```
vector <int> v = {1, 2, 3, 4};
for (auto i = 1; i <= v.size()-1; i++){
    std::cout << v[i] << ", ";
}
```

i. prints 1, 2, 3, 4

ii. prints 2, 3, 4,

iii. throws an error

iv. undefined

(d) If b is a variable of the class Ball (below), how would you call the function draw? [4]

```
class Ball{
private:
   int radius;
public:
  Ball(int r);

    void draw(int x, int y);
}

Ball b(6);
```

i. `b.draw(10, 20);`

ii. `b->draw(10, 20);`

iii. `Ball.draw(10, 20);`

iv. `Ball->draw(10, 20);`

(e) You are trying to print the numbers from 0 to 10 in reverse order but this code does not work. What should you change? [4]

```
  for (int i = 10; i >= 0; i++){
std::cout << "number" << i << ", ";
  }
```

i. you should replace `int` with `auto`

ii. you should convert i to a string, `cout` can't print `ints`

iii. change `i++` to `i--`

iv. change `i >= 0` to `i > 0`

(f) What is wrong with this function? [4]

```
#include <memory>

void fun(){
    std::unique_ptr<int> p(new int(3));
    std::cout << *p;
}
```

i. using it could or will cause a compile-time or run-time error

ii. using it could or will cause undefined behaviour

iii. using it could or will cause a memory leak

iv. there is nothing wrong with this function

(g) Which of the following must be included in the header file to perform file I/O operations? [4]

i. `<fstream>`

ii. `<ofstream>`

iii. `<ifstream>`

iv. any of the above

(h) When are default values for a function specified? [4]

i. when a function is defined

ii. when a function is declared

iii. both i and ii

iv. none of the above

(i) Which of the following is/are used for generic programming? [4]

i. inheritance

ii. virtual functions

iii. templates

iv. none of the above

(j) The derivation of a child class from a base class is indicated by which symbol? [4]

i. ::

ii. ;

iii. :

iv. –

Part B

Question 3 Data structures

(a) java.util.ArrayList is a container that is part of the Java Collections Framework. Name three other container types that are part of the Java Collections Framework. [3]

(b) What are the advantages of using the java.util.ArrayList container type? [3]

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

```
import java.util.*;
public class PAP8c {

       static void function(ArrayList<String> s) {
              ArrayList<String> words = new ArrayList<String>();
              Iterator<String> iter = s.iterator();

              while (iter.hasNext()) {
                     words.add(iter.next());
              }

              String result = "";
              while (!words.isEmpty()) {
                     result = result + " " + words.get(words.size() - 1);
                     words.remove(words.size() - 1);
              }
              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");
              function(s);
       }
}
```

[8]

(d) What are exceptions used for? [2]

(e) The following code terminates with an uncaught exception of type java.lang.IndexOutOfBoundsException:

```
import java.util.ArrayList;
public class PAP8e {

    public static void main(String[] args) {

        ArrayList<Integer> v = new ArrayList<Integer>();

        for (int i = 0; i < 10; ++i) {
            v.add(i);
        }
        v.get(10);
        System.exit(0);
    }
}
```

i. Describe what the code does [5]

ii. Using exception handling rewrite the above code so that "Error" is printed to the console if a user tries to access a location outside of the ArrayList v. [9]

Question 4 Inheritance

(a) What is the difference between interfaces and abstract classes ? [4]

(b) Given the following definition:

```
interface Mammal{
     abstract boolean isPlacental();
     abstract boolean isCarnivore();
}
```

Say which of the following classes will compile successfully and which will not.

```
abstract class Omnivore implements Mammal{
     abstract boolean isScavenger();
}

class Marsupial implements Mammal{
     public boolean isPlacental(){
          return false;
     }
}

class Rabbit implements Mammal{
     public boolean isCarnivore(){
          return false;
     }
     public boolean isPlacental(){
          return true;
     }
}
```

[6]

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

- a volunteer (`Volunteer`) is-a staff member (`StaffMember`)
- an employee (`Employee`) is-a staff member (`StaffMember`)
- an executive (`Executive`) is-an employee (`Employee`)
- an hourly paid employee (`HourlyPaid`) is-an employee (`Employee`)

The `StaffMember` and `Volunteer` classes are given as follows

```
// %%%%%%%%%%%%%%%%%%%% StaffMember class %%%%%%%%%%%%%%%%%%
abstract class StaffMember {
   protected String name;
   protected String address;
   protected String phone;
   public StaffMember (String name, String address, String phone)   {
      this.name = name;
      this.address = address;
      this.phone = phone;
   }
   public String toString ()   {
      String result = "Name: " + name + "\n";
      result += "Address: " + address + "Phone: " + phone;
      return result;
   }
   public abstract double pay();
}

// %%%%%%%%%%%%%%%%%%  Volunteer class   %%%%%%%%%%%%%%%%%%%%%
class Volunteer extends StaffMember {
   public Volunteer (String name, String address, String phone)   {
      super (name, address, phone);
   }
   public double pay()   {
      return 0.0;
   }
}
```

(i) Give a Java class definition for a class Employee that has two more attributes

- protected String socialSecurityNumber; // identifies the employee's social security number
- protected double payRate; // identifies the employee's pay rate
- The pay function should use the pay rate

[6]

(ii) Give a Java class definition for a class Executive with following

- One more attribute: private double bonus; // identifies the bonus paid
- A member function `void awardBonus()` that sets the bonus
- The `pay()` function has to take into account the bonus given to the executive

[6]

(ii) Give a Java class definition for a class `Hourly` with following

- One more attribute: private int hoursWorked; // identifies the number of hours worked
- A member function `void addHours (int moreHours)` that updates the number of hours worked
- The function `pay()` should now return the value ‘payRate*hoursWorked’
- The function `toString()` should now include additional information about hours worked

[8]

Question 5 JDBC

(a) What are the basic steps when using JDBC? [6]

(b) Given the following program ‘Jdbc2017.java’:

```
import java.io.*;
import java.sql.*;
import java.util.*;
public class Jdbc2017{
  public static void main(String[]args){
      String  driver = "com.mysql.jdbc.Driver",
                  url = "jdbc:mysql://igor.gold.ac.uk",
                  database = "mas01lo_java",
                  user = "xxxxxxx",
                  password = "yyyyyy",
                  student  = "sudent_table";
      try{
          Class.forName(driver);
           try{
               Connection db = DriverManager.getConnection(url + "/" + database,
                                                         user, password);
                try{
                    db.createStatement().executeUpdate("CREATE TABLE " + student +
                " (id int PRIMARY KEY, name text,course text,  department text)");

                  try{
                      FileInputStream fstream = new FileInputStream("student.txt");

                 //   %%%%%%%%   add a code to  part (c) here   %%%%%%%%
                        }
                          //   %%%%%%%%   add a code to  part (d) here   %%%%%%%%

                       catch(Exception e){System.out.println("first error");}
                    }

                   catch(Exception e) {System.out.println("second error");}
             }
               catch(Exception e) {System.out.println("third error");}
        }
         catch(Exception e){System.out.println("fourth error");}
       }
     }
}
```

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

ii. What would be the output if the file ‘student.txt’ was not found?

[8]

(c) Add the missing code fragment into the program to store each line of the file ‘student.txt’ in an ArrayList called ‘list’.

The file ‘student.txt’ contains the following:

```
id,    name,           course,    department
1,    Iqbal Nauman,       Computer Science,  Computing
2,    Bader Adam,    Game Programming,   Computing
```

[8]

(d) Add the missing code fragment to insert each line in the file ‘student.txt’ as an arrow in the student table. [8]

Part C

Question 6 Memory and Pass-by-Reference

(a) Explain what a memory leak is. [4]

(b) Describe two ways of avoiding memory leaks. [4]

(c) Explain what “pass by const reference” is. [5]

(d) Give example code of a function that uses “pass by const reference” for an object parameter and one that uses “pass by value”. Give examples of calling each function. [7]

(e) Explain what memory is allocated on the heap, what memory is allocated on the stack and what memory is copied, when the following function is called. [8]

```
int f(int &a){
  int *b = &a;
  int *c = new int;
  *c = *b;
  return *c;
}
```

(f) What error does the code in the previous part contain [2]

Question 7 Inheritance

(a) In C++, what is an abstract class? In what contexts would you use an abstract class (give an example)? [4]

(b) Given the following class definitions. Which of the variable definitions below will compile?

```
class Mammal {
public:
    virtual bool isPlacental()=0;
    virtual bool isCarnivore()=0;
};

/*********************************/
class Omnivore : public Mammal {
public:
    virtual bool isScavenger() = 0;
};

/*********************************/
class Marsupial : public Mammal {
public:
    bool isPlacental();
};

bool Marsupial::isPlacental(){
    return false;
}

/*********************************/
class Rabbit : public Mammal {
public:
    bool isCarnivore();
    bool isPlacental();
};

bool Rabbit::isCarnivore() {
    return false;
}
bool Rabbit::isPlacental() {
    return true;
}
```

```
/**********************************/
class Bear : public Omnivore {
public:
    bool isCarnivore();
    bool isPlacental();
};

bool Bear::isCarnivore() {
    return true;
}
bool Bear::isPlacental() {
    return true;
}

/**********************************/
class Kangaroo : public Marsupial {
public:
    bool isCarnivore();
};

bool Kangaroo::isCarnivore() {
    return false;
}
```

i. `Mammal m;`

ii. `Omnivore o;`

iii. `Marsupial m;`

iv. `Rabbit r;`

v. `Bear b;`

vi. `Kangaroo k;`

[6]

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

- a volunteer (`Volunteer`) is-a staff member (`StaffMember`)
- an employee (`Employee`) is-a staff member (`StaffMember`)
- an executive (`Executive`) is-an employee (`Employee`)
- an hourly paid employee (`HourlyPaid`) is-an employee (`Employee`)

The `StaffMember` and `Volunteer` classes are given as follows:

```
// %%%%%%%%%%%%%%%%%%%% StaffMember class %%%%%%%%%%%%%%%%%
```

```
class StaffMember {
protected:
    std::string name;
    std::string address;
    std::string phone;
public:
    StaffMember (std::string name, std::string address, std::string phone);
    virtual std::string toString ();
    virtual double pay() = 0;
};

StaffMember::StaffMember (std::string _name,
          std::string _address, std::string _phone)
          : name(_name), address(_address), phone(_phone) {
}

std::string StaffMember::toString () {
    std::string result = "Name: " + name + "\n";
    result += "Address: " + address + "\nPhone: " + phone;
    return result;
}

// %%%%%%%%%%%%%%%%%% Volunteer class %%%%%%%%%%%%%%%%%%%%%

class Volunteer : public  StaffMember {
public:
    Volunteer (std::string name, std::string address, std::string phone);
    double pay();
};

Volunteer::Volunteer (std::string name,
           std::string address, std::string phone)
           :StaffMember (name, address, phone) {
}

double Volunteer::pay() {
    return 0.0;
}
```

(i) Give a C++ class definition for a class Employee that has two more protected member variables

- std::string socialSecurityNumber; // identifies the employee's social security number
- double payRate; // identifies the employee's pay rate

- The pay function should use the pay rate

[6]

(ii) Give a C++ class definition for a class Executive with the following

- One more member variable: double bonus; // identifies the bonus paid
- A member function `void awardBonus()` that sets the bonus
- The `pay()` function has to take into account the bonus given to the executive

[6]

(ii) Give a C++ class definition for a class `Hourly` with the following

- One more member variable: int hoursWorked; // identifies the number of hours worked
- A member function `void addHours (int moreHours)` that updates the number of hours worked
- The function `pay()` should now return the value 'payRate*hoursWorked'
- The function `toString()` should now include additional information about hours worked

[8]

Question 8 std::vector & Exceptions

(a) `std::vector` is a container that is part of the standard library. Name three other container types that are part of the standard library. [3]

(b) What are the advantages of using the `std::vector` container type? [3]

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

```
#include <iostream>
#include <string>
#include <vector>

   void function(std::vector<std::string> s) {
       std::vector<std::string> words;
       auto iter = s.begin();
       while (iter != s.end()){
           words.push_back(*iter);
           iter++;
       }
       std::string result = "";
       while (!words.empty()) {
           result = result + " " + words.back();
           words.erase(words.end()-1);
       }
       std::cout << result << std::endl;
   }
   int main(int argc, const char * argv[]) {
       std::vector<std::string> s;
       s.push_back("here");
       s.push_back("comes");
       s.push_back("summer");
       function(s);
   }
```

[8]

(d) What are exceptions used for? [2]

(e) The following code terminates with an uncaught exception of type `std::out_of_range`

```
#include <vector>

int main ()
{
    std::vector<int> v;

    for (int i = 0; i < 10; i++){
        v.push_back(i);
    }

    v.at(10);

    return 0;
}
```

(f) Describe what the code in part (e) does [5]

Using exception handling rewrite the above code so that "Error" is printed to the console if a user tries to access a location outside of the vector v. [9]