UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2019

IS52028A
Principles and Applications of Programming

Duration: 2 hours 15 mins

Date and time:

This paper is in THREE parts: part A, part B and part C. You should answer ONE question from part A (either C++ or Java) and TWO questions from part B (Java) or part C (C++). Part A carries 40 marks, and each question from part 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 for this paper.

Electronic calculators must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.

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

Part A

Multiple choice

Question 1 Multiple choice JAVA

(a) What is the output of the following programs:

```

class Test {
    protected int x,
    public int y;
}

class Main {
    public static void main(String args[]) {
        Test t = new Test();
        System.out.println(t.x + "␣" + t.y);
    }
}
```

[2]

i. 0 0

ii. null 0

iii. null null

iv. compilation error as the variable x is protected and hence, it isn't access from the Main class

(b) What is the output of the following code snippet:

```

public static void main(String args[]) {
     try {
              int x, y;
              y = 0;
              x = 1/y;
              System.out.print("X␣");
         }

     catch(ArithmeticException e) {
                    System.out.print("Y␣");
       }

     finally {
                  System.out.println("finally");
     }
 }
```

[2]

i. X

ii. Y

iii. X finally

iv. Y finally

(c) The order of execution of constructors in Java Inheritance is [2]

i. base to derived
ii. derived to base
iii. random order
iv. none of the above

(d) Which one statement is correct about the following code snippet? [2]

```
double  x =2 ;
int y = x
System.out.println(x);
```

i. it prints the 2
ii. it prints 2.0
iii. compilation error
iv. runtime error

(e) What is the value of x at the end of funct and why? [2]

```
 void funct(){
   int x = 10;
   double  y = 20.5 ;
   x ++;
   x = x + y;
}
```

i. 32 because y is rounded up when it is added to x.
ii. 31 because y is rounded down when it is added to x.
iii. This code will not compile as it assigns a number value to an int.
iv. 30 because y is rounded down when it is added to x.

(f) Which of the following statements are **INCORRECT**? [2]

i. static methods can call other static methods only
ii. static methods must only access static data
iii. static methods can not refer to this or super in any way
iv. when an object of a class is declared, it contains its own copy of static variables

(g) Which of these classes implements Set interface?. [2]

i. List
ii. ArrayList
iii. LinkedList
iv. HashSet

(h) Which among the following is **CORRECT**? [2]

i. class Student{ public Student(){} }
ii. lass Student{ public int Student(){} }
iii. class Student{ public void Student (){} }
iv. class Student{ public Student{}{}}

(i) Which interface restricts duplicate elements?. [2]

i. List
ii. Map
iii. Set
iv. none of the above

(j) In which Java **OOP** feature can one object acquire all the properties and behaviours of the parent object? [2]

i. inheritance
ii. polymorphism
iii. encapsulation
iv. abstraction

(k) Which inheritance in java programming is not supported? [2]

i. Multiple inheritance using interfaces
ii. Multiple inheritance using classes
iii. Multilevel inheritance
iv. Single inheritance

(l) Which one of the following correctly describe polymorphism? [2]

i. It is when a single variable is used with several different types of related objects at different places in a program
ii. It is when a program uses several different types of objects, each with its own variable
iii. It is when a single parent class has many child classes
iv. It is when a class has several methods with the same name but different parameter types

(m) Which class cannot be sub classed? [2]

i. final class
ii. abstract class
iii. object class
iv. child class

(n) If private members are to be called outside the class, which is a good alternative? [2]

i. Add a private member function which calls the private member function.
ii. Add a public member function which calls the private member function.
iii. Add a protected member function which calls the private member function.
iv. Not possible.

(o) Given the following code snippet: [2]

```
abstract class Student{
   public int marks;
   abstract double calc_grade();
}

class Topper extends Student{
    public double  calc_grade() {
        return 10;
    }
}

class Average extends Student{
     public  double calc_grade(){
         return 20;
     }
}

class Failed{ int marks; }
```

Which one of the following statements is **correct**?

i. only class Student can show polymorphism

ii. only class Student and Topper together can show polymorphism

iii. class Student, Topper and Average together can show polymorphism

iv. class Failed should also inherit class Student for this code to work for polymorphism

(p) Given the following method:

```
private int f( int x, int y ) {
  if( x == 0 ) {
      return y;
  }
  else {
      return f( x - 1, y + 1 );
   }
}
```

What is the value of of f(8,9)? [2]

i. 0

ii. 17

iii. this recursion is incorrect

iv. 72

(q) What type of relationship exists between **someMeth** in classes A and **someMeth** in class B? [2]

```
class A{
    private void someMeth(){
       System.out.println( "from␣class␣A" );
    }
}

class B extends A{
    public void someMeth( String x ){
        System.out.println( "from␣class␣B:␣" + x );
}
}
```

i. method overriding

ii. method overloading

iii. both method overriding and method overloading

iv. neither method overriding nor method overloading

(r) Given the following definition of Bird and Chicken, which of the given statements will not compile? [2]

```
abstract class Bird implements Livestock {}
class Chicken extends Bird {}
```

i. Bird bird = new Chicken();

ii. Livestock livestock = new Chicken();

iii. Bird bird = new Bird();

iv. none of the above

(s) What is the output of this program? [2]

```
class A{
    public int i;
    private int j;
}

class B extends A{
    void display(){
          super.j = super.i + 1;
          System.out.println(super.i + "␣" + super.j);
    }
}

class inheritance{
    public static void main(String args[]){
          B obj = new B();
          obj.i=1;
          obj.j=2;
          obj.display();
    }
}
```

i. 22

ii. 33

iii. run time error

iv. compilation error

(t) What will be the result of compiling and running the following code? [2]

```
class A{
     int b=10;
     A(){
      this.b=7;
     }
     int f(){
       return b;
     }
}

class B extends A{
     int b;
     B(int b){
       this.b=b;
    }
}

public class Test10{
      public static void main(String[] args){
            A a = new B();
            System.out.println(a.f());
      }
}
```

i. compilation error

ii. prints 0

iii. prints 7

iv. prints 10

Question 2 Multiple choice C++

(a) You need a one dimensional array that can store the data that represents the contents of the screen. The screen resolution is 3840 x 2160 pixels and each pixel is represented with four eight bit channels. Which array declaration(s) is appropriate for storing this data and will store all the data in as little memory as possible, assuming char is 8 bit, unsigned int is 32 bit and boolean is 8 bit? Select all that apply. [2]

i. bool pixels[8294400]

ii. unsigned int pixels[8294400]

iii. char pixels[33177600]

iv. char pixels[265420800]

(b) What is the correct code to multiply the stored values in the vector by ten? Select all the correct options. [2]

i.

```
std::vector<int> vals = {2,4,6,8};

for (int i=0;i<vals.size();i++){
  vals[i] = vals[i] * 10;
}
```

ii.

```
std::vector<int> vals = {2,4,6,8};

for (int i : vals){
    i ++ 10;
}
```

iii.

```

std::vector<int> vals = {2,4,6,8};

for (int i : vals){
    vals[i] *= 10 ;
}
```

iv.

```

std::vector<int> vals = {2, 4, 6, 8};

for (int i =0; i<vals.size(); i++){
    vals.at(i) *= 10 ;
}
```

(c) Which are true about the code below? Select all that apply.

```
#include <iostream>

void calltheplumber(){
  float* data = new float[10];
  data[0] = 100.0;
  std::cout << "data[0]␣" << data[0] << std::endl;
}

int main(){
  calltheplumber();
}
```

[2]

i. calltheplumber contains a memory leak as the word new is used without a matching delete call.

ii. Adding the line delete data; at the end of calltheplumber would fix the leak, if there is one.

iii. Adding the line delete data; at the end of main would fix the leak, if there is one.

iv. data points to a block of memory that can only store a single float value.

(d) You see the following code:

```
unsigned char values[1000];
```

Which of the following are true (select all that apply)? [2]

i. The array can store values in the range 0-511 if chars are 8 bit.

ii. The array will be allocated on the stack.

iii. The array will be allocated in the heap.

iv. The array will start taking up memory once this line has been executed.

(e) You see a function declaration like this:

```
int findTheHighest(std::vector <int> values);
```

Why is this function going to be inefficient? Select one answer.

[2]

i. The name of the function is too long so it will take longer to compile than a function with a shorter name would.

ii. It uses pass by reference so it will make a copy of the vector when it is called.

iii. It uses pass by value so it will make a copy of the vector when it is called.

iv. We don't know if it is inefficient as we cannot see the implementation.

(f) Consider the follow code. How many times is the destructor Car called?

```
#include <iostream>

class Car {
   public:
     Car();
     ~Car();
     void goFaster();
     void goSlower();
   private:
     float speed;
};

Car::Car() :speed{0}{
}
Car::~Car(){
   std::cout << "car␣dest" << std::endl;
}
void Car::goFaster(){
   speed ++;
}
void Car::goSlower(){
   speed --;
}
void speedUpMyCar(Car &car){
   car.goFaster();
}
void slowDownMyCar(Car car){
   car.goSlower();
}
int main(){
   Car car1;
   speedUpMyCar(car1);
   slowDownMyCar(car1);
}
```

[2]

i. 3

ii. 0

iii. 2

iv. 4

(g) Consider the following code. When will the destructor ˜ Car be called (select all that apply)?

```
#include <iostream>

class Car {
  public:
    Car();
    ~Car();
    void goFaster();
    void goSlower();
  private:
    float speed;
};

Car::Car() :speed{0}{
}
Car::~Car(){
  std::cout << "car dest" << std::endl;
}
void Car::goFaster(){
  speed ++;
}
void Car::goSlower(){
  speed --;
}
void speedUpMyCar(Car &car){
  car.goFaster();
}
void slowDownMyCar(Car car){
  car.goSlower();
}
int main(){
  Car car1;
  speedUpMyCar(car1);
  slowDownMyCar(car1);
}
```

[2]

i. At the end of the constructor.

ii. At the end of speedUpMyCar.

iii. At the end of slowDownMyCar.

iv. At the end of main.

(h) A class has a constructor like this:

```
Ball(float xPos, float yPos);
```

Which of the following lines will call this constructor? Select all that apply.

[2]

i. `Ball* ball1 = new Ball(10, 20);`

ii. `Ball ball2{10, 10};`

iii. `Ball ball4[10, 20];`

iv. `Ball ball3(10, 20);`

(i) What does the following program do? Select one option.

```
#include <iostream>

int main()
{
  int i = 0;
  while (i<5) {
     std::cout << i << ":␣" << (i*2) << std::endl;
     i++;
  }
}
```

[2]

i. It prints the numbers from 0 to 4 inclusive and the squares of these numbers.

ii. It prints the numbers from 0 to 4 inclusive and these numbers scaled by 2.

iii. It prints the numbers from 0 to 5 inclusive and these numbers scaled by 2.

iv. It does not run as there is an infinite loop.

(j) Which of the following are true about pointers? Select all that apply. [2]

i. Pointers allow us to allocate memory on the stack.
ii. Pointers allow us to allocate memory on the heap.
iii. Pointers do not take up any memory until we call new.
iv. Pointers are the same as references.

(k) In the following code, the doSomethingToABall function receives a pointer to a ball. However, in main, ball is not a pointer, yet it can be passed in. Why is that?

```
class Ball{
   public:
    Ball(int x, int y);
};
Ball::Ball(int x, int y){
}

void doSomethingToABall(Ball* ball){
}

int main(){
   Ball ball {10,10};
   doSomethingToABall(&ball);
}
```

[2]

i. Ball and Ball* variables are interchangeable in C++.

ii. The ampersand character & allows you to access the data inside the ball variable, even if it is private.

iii. The ball variable is being passed by value which is equivalent to passing in a pointer.

iv. The ampersand character & allows you to access the memory address of of the ball variable, and a pointer is a memory address.

(l) Which of the following are the correct statements to print out the amount of memory taken up by an array of integers declared like this:

```
int data[16];
```

Select all that apply [2]

i. `std::cout << sizeof(int) << std::endl;`

ii. `std::cout << (sizeof(int) * 16) << std::endl;`

iii. `std::cout << (sizeof(data) * 16) << std::endl;`

iv. `std::cout << (sizeof(data) * sizeof(int)) << std::endl;`

(m) What is true about virtual functions? Select all that are true. [2]

i. The implementation of a virtual function that is called is decided based on the run-time type of the object.

ii. Virtual functions allow for polymorphism, where objects can treated as different classes.

iii. If a parent class defines a function as virtual, the parent implementation will never be called, it will always be the implementation from a sub class that is called.

iv. Pure virtual functions do not have any implementation.

(n) Consider the following code:

```
class Shape{
  public:
   void draw();
};
class Box : public Shape {
  public:
   void draw();
};
void Shape::draw(){
}
void Box::draw(){
}
int main(){
  Shape* shape = new Box();
  shape->draw();
}
```

Which version of draw will be run? [2]

i. Shape::draw only.

ii. Box::draw only.

iii. Neither.

iv. Both.

(o) Consider the following code:

```
class Shape{
  public:
   virtual void draw();
};
class Box : public Shape {
  public:
   virtual void draw();
};
void Shape::draw(){
}
void Box::draw(){
}

int main(){
  Shape* shape = new Box();
  shape->draw();
}
```

Which version of draw will be run? [2]

i. Shape::draw only.

ii. Box::draw only.

iii. Neither.

iv. Both.

(p) Consider the following code. Which constructor is called?

```
class Shape{
  public:
   Shape();
};
class Box : public Shape {
  public:
   Box();
};
Shape::Shape(){
}
Box::Box(){
}
int main(){
  Shape* shape = new Box();
}
```

[2]

i. Box::Box only.

ii. Shape::Shape only.

iii. Neither.

iv. Both.

(q) Select the most accurate definition of compounding. [2]

i. A class inheriting **public** data and function members from a parent class.
ii. A class using other classes as data members.
iii. A class inheriting **private** data and function members from a parent class.
iv. A class implementing virtual functions inherited from a parent class.

(r) Consider the following code. Which of the following programming techniques are being used? Select all that apply.

```
class PositionVector{
   public:
     void setPosition(float _x, float _y, float _z);
   private:
    float x;
    float y;
    float z;
};
class Shape {
  public:
    void setPosition(float _x, float _y, float _z);
  private:
    PositionVector pos;
};
class Cube : public Shape {
  public:
};
void PositionVector::setPosition(float _x, float _y, float _z
    ){
  x = _x;
  y = _y;
  z = _z;
}
void Shape::setPosition(float _x, float _y, float _z){
  pos.setPosition(_x, _y, _z);
}
int main(){
   Cube cube;
   cube.setPosition(10, 100, 5);
}
```

[2]

i. Compounding

ii. Polymorphism

iii. Inheritance

iv. Overloading

(s) Consider the following code. What kind of thing is x?

```
int y = 10;
int z = y;
int& x = y;
```

[2]

i. an int

ii. a reference to an int

iii. a pointer to an int

iv. an unsigned int

(t) Consider the following code. What kind of thing is y?

```
int x = 10;
int z = x;
int* y = &x;
```

[2]

i. an int

ii. a reference to an int

iii. a pointer to an int

iv. an unsigned int

Part B

Written answer questions JAVA

Question 3 Data Structure – JAVA

(a) Given the following code snipet:

```
public static Queue<Integer> mystery(Queue<Integer> q) {
                Queue<Integer> temp = new LinkedList<Integer>();
                while (!q.isEmpty()) {
                        int n = q.poll();
                        temp.add(n);
                        temp.add(n);
                }
                return temp;
        }
}
```

What will be printed if the following code is executed?

```
Queue<Integer> q = new LinkedList<Integer>();
q.add(1);
q.add(2);
q.add(3);
System.out.println(mystery(q));
}
```

[6]

(b) Given the following unfinished program:

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

public class FileManipulation {
  public static ArrayList<String> fileToArrayList(String file) throws IOException{

        String filename= path+"/"+file;
        Scanner scan = new Scanner(new File(filename));
        ArrayList<String> list=new ArrayList<String>();
        while (scan.hasNextLine()) {
                String line = scan.nextLine();
                String delim = " ,:?!/[].;\t\n";
                StringTokenizer st = new StringTokenizer(line, delim);

                // add your code here!

        }




        public static Map<String,Integer> fileToMap(String file) throws IOException
        {
                // add your code here
        }

        public static void MostFrequentWords(String file) throws IOException{
                //add your code here
        }
}

}
```

i. Finish the implementation of the function **fileToArrayList**. This function take a filename as its argument and returns an ArrayList containing all the words in the file. [8]

ii. Implement **fileToMap**. This function takes a filename as its argument and returns a HashMap containing the words in the file and the number of their occurrences.MostFrequentWords **fileToArrayList** . [8]

iii. Implement **MostFrequentWords**. This function takes a filename as its argument and prints the most frequently used word in the file in its the number of its occurrences . This function should contain a call to function **fileToMap**.

[8]

Question 4 Inheritance & Polymorphism

(a) Explain the difference between an interface and an abstract class. [6]

item

i. Describe three modifiers which can be used to define the visibility of class members (fields and methods).

ii. Explain why it is considered good practice to limit the scope of fields and methods in object oriented programming.

[6]

(b) Consider the following code which define a class:

```
 1  abstract class Animal extends  LivingThing {
 2       ArrayList <String> toDoList = new ArrayList<>();
 3       public Animal(){ }
 4       protected void digestFood(){}
 5       protected bool doYouWantToDoThis(String description) {
 6            for (String e: toDoList){
 7             if(e.equals(descriptions))
 8                  return true;
 9            }
10            return false;
11       }
12  }
```

For each of the following items, identify on which line numbers, or line number ranges they occur in the code, and explain why you selected the line number(s):

i. A set of functions that can be called by subclasses of Animal but not by classes which have Animal objects as data members. [2]

ii. The constructor. [2]

iii. The definition of the inheritance relationship. [2]

iv. The method that returns a value. [2]

(c) You want to add a public function to the Animal class called eat. It takes a Food object as an argument and it does not return anything. It should be able to be used polymorphically so that subclasses of Animal can implement their own version. Write the function prototype for the eat function. [4]

(d) You have been asked to create a simulation of a zoo for a kids' video game. Write three classes (Tiger, ZooKeeper, and SerbianTiger). Include constructors, data members and function members. Your classes and their members have to illustrate the following relationships:

i. Inheritance

ii. Polymorphism

State which class demonstrates which relationship. The classes might show more than one relationship. [6]

Question 5 Methods & Exceptions

(a) i. Which Java class is the parent class of all Exceptions?

ii. What is the difference between Exceptions and Errors?

[8]

(b) Change the divide method so that it throws an IllegalArgumentException with an appropriate message if b is zero.

```
public static double divide(double a, double b) {
 return a / b;
}
```

[4]

(c) Given the following code snippet:

```
public static void mystery(int [] array) {
        int tmp = array[array.length - 1];
        for (int i = 1; i < array.length; i++) {
                array[i] = array[i - 1];
        }
        array[0] = tmp;
   }
}
```

What is printed when the following code is executed?

```
int[] list = { 10, 20, 30, 40, 50, 60};
mystery(list);
for(int i=0; i<list.length;i++){
    System.out.print(list[i] + "␣") ;
}
}
```

[4]

(d) Given the following method:

```
static boolean mystery(int[] list)
        {
          Set<Integer> set = new HashSet<Integer>();
          for (int i : list)
          {
            if (set.contains(i)) return true;
            set.add(i);
          }
          return false;
        }
```

i. Explain what does the method **mystery** do.

ii. What is the value of **mystery(arr);** is array is an array containing the following values 1, 2, 3, 5, 3, 6 ?

[6]

(e) Write a method called **isPalindrome** that takes a string as its argument, returns true if the the string is a palindrome and false otherwise. [8]

Part C

Written answer questions C++

Question 6 Interactions between classes

(a) Consider the following code which defines the prototype for a class:

```
1  class Animal : public LivingThing {
2     public:
3       Animal();
4     protected:
5       void digestFood();
6       bool doYouWantToDoThis(std::string description);
7  };
```

For each of the following items, identify on which line numbers, or line number ranges they occur in the code, and explain why you selected the line number(s):

i. A set of functions that can be called by subclasses of animal but not by classes which have Animal objects as data members. [2]

ii. The constructor. [2]

iii. The definition of the inheritance relationship. [2]

iv. The prototype of a function that returns a value. [2]

(b) This code would normally go in a header file. Write the implementation code that would be found in the cpp file for the Animal class. You can leave the function bodies empty. [6]

(c) You want to add a public function to the Animal class called eat. It takes as its argument a pointer to a Food object and it does not return anything. It should be able to be used polymorphically so that subclasses of Animal can implement their own version. Write the function prototype for the eat function. [4]

(d) You have been asked to create a simulation of a zoo for a kids' video game. Write the class prototypes for 4 or more classes. Include constructors, data members and function members. Choose your classes and their members such that they illustrate the following relationships:

i. Inheritance

ii. Polymorphism

iii. Compounding

iv. Dynamic memory allocation

State which class demonstrates which relationship. The classes might show more than one relationship. [8]

(e) Explain how you would use the features of the openframeworks library to draw and animate a simple animal in the zoo game. [4]

Question 7 Testing and handling unexpected values

(a) You are working on the testing for a mathematical library. There is a function with the following prototype:

```
int pow(int x, int y);
```

Here are two requirements:

- It correctly computes powers of values of x in the range 2-10 and y in the range 1-10.
- It returns -1 if negative nunbers are passed in

i. Explain how you would go about testing each of these requirements using a set of unit tests. [4]

ii. Write the code for a test for each requirement. [6]

(b) What is exception handling? Explain how it works. [3]

(c) Write some code that uses exception handling to deal with values that are not in the specified ranges for x and y, as per the requirements above. You do not need to implement the complete pow function yet, just the part that checks the incoming values. Show how the exception handling works in the called and calling functions, by writing an example main function that calls pow and handles the exception. [4]

(d) Where is the best place in the code to decide what to do if unexpected values are sent into a function? In the function or in the code that calls the function? Why? [3]

(e) Write an implementation for the pow function. Lay the code out neatly and write comments explaining what is going on. You do not need to include the code that you wrote previously to deal with unexpected values. [6]

(f) Discuss the difference between unit testing and exception handling. When in the lifetime of the program from development to deployment would each technique be used? [4]

Question 8 Interactions between classes

You are creating a video game in the 'Battle Royale' genre. In these games, a group of players are dropped on an island with equipment and weapons scattered around, then they have to eliminate all other players, to become the 'last person standing'. Fortnite is an example of such a game. In this question, you will design various aspects of the game.

(a) Write a class prototype for a base Weapon class. Write a subclass prototype for a Catapult class. Weapons should have the following characteristics:

i. A Weapon should be able to be used a limited number of times. [2]

ii. Subclasses of Weapon should be able to configure the number of times they can be fired using a function member that is only accessible to subclasses of Weapon. [2]

iii. A Weapon should cause a certain amount of damage, again, configurable only by subclasses of Weapon, and not from non-subclasses. [2]

iv. The game should be able to query a weapon for how many shots it has left via a function.

[4]

(b) Write a class prototype for a base ProtectiveItem class which can protect a player from weapons. Think about how the protection might decay under attack from a weapon, making use of your Weapon class where appropriate. [4]

(c) Write a class prototype for a Player class. It should have the following characteristics:

i. Plyaers have a set of protective items and a set of weapons, stored in vectors. [2]

ii. Players have a current weapon and a current protective item. [2]

iii. Players are able to be attacked with a weapon, and to provide access to their current weapon. [2]

(d) Some objects in the game can have two purposes - for example, you can hide under a table (ProtectiveItem) and you can throw it at someone (Weapon).

i. Which object oriented techniques would you need to use in combination to allow a class to present itself differently to different parts of the program, in both cases with a limited 'interface'? [2]

ii. Draw a class diagram showing the inheritance pattern you would need to be able to control the interface a class presents to various other classes, to enable a Table class to operate as a Weapon and a ProtectiveItem. List the interactions your diagram enables and how the interface netween interacting classes is controlled. [4]

(e) There is a World class which contains all objects - players, weapons and protective items. Describe in terms of class interactions and the function and data members you have in your Player class above, how player 1 would go about hiding under a table, whilst player 2 attacked them with an axe, then player 1 picks up the table and attacks player 2 with it. Comment on how the state of the various objects involved would change. [4]