

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2018-Resit

IS52028A

Principles and Applications of Programming

Duration: 2 hours 15 minutes

Date and time:

This paper is in two parts: part A and part B. You should answer ALL questions from part A and TWO questions from part B. Part A carries 40 marks, and each question from part B 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.

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) Considering the following code, which one of the following is true?

```
1
2 class Bike {
3     private int speed;
4
5     Bike (){
6         speed = 0;
7     }
8     Bike (double n){
9         new Bike();
10    }
11    public static void main(){
12        Bike bike = new Bike(10);
13    }
14 }
```

[2]

- i. The variable speed is initialised to zero in the constructor.
- ii. The variable speed is initialised to null in the constructor.
- iii. The variable speed is initialised to 10 in the constructor.
- iv. The variable speed can be directly accessed from any sub-class of the Car class.

- (b) Which is the correct value and explanation for the value of the colour variable inside the bike object at the end of main? [2]

```
1
2 class MotorBike{
3     private String colour;
4
5     MotorBike(String c){
6         colour = c;;
7     }
8
9     void setColour(String c){
10        colour = c;
11    }
12
13    static void respray(Motorbike bike){
14        bike.setColour("green");
15    }
16
17    public static void main(String args[]){
18        MotorBike bike = new MotorBike("red");
19        respray(bike);
20        System.out.println(bike.colour);
21    }
22 }
```

- i. It is red because the respray function implements pass by reference, not pass by value.
- ii. It is red because the respray function implements pass by value, not pass by reference.
- iii. It is green because the respray function operates upon the bike object, setting its class member colour to a new value.
- iv. It is not possible to know what the colour would be as the bike variable is not returned from the respray function.

(c) Which one the following is a valid call to the constructor of class Bicycle? [2]

- i. Bicycle bike;
- ii. Bicycle bike = new Bicycle();
- iii. Bicycle new Bicycle;
- iv. None of the above

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

```
String x ;  
x = 10;  
System.out.println(x);
```

- i. It prints the string "10"
- ii. It prints the integer 10
- iii. Compilation error
- iv. None of the above

(e) Which one is true about the following code? [2]

```
int x = 5;  
double y = 10.5 ;  
x ++;  
y = x + y;  
System.out.println(y);
```

- i. It prints 17 because y is rounded up when it is added to x.
- ii. It prints 16.5 because adding a double to int result in a double value.
- iii. This code will not compile as it assigns a double value to an int.
- iv. None of the above

(f) Which one is true about the following code? [2]

```
int x = 5;
double y = 10.5 ;
x ++;
x= x + y;
System.out.println(x);
```

- i. It prints 17 because y is rounded up when it is added to x.
- ii. It prints 16.5 because adding a double to int result in a double value.
- iii. This code will not compile as it assigns a double value to an int.
- iv. None of the above

(g) What is the difference between an ArrayList data structure and an array data structure? [2]

- i. An ArrayList is a fixed sized data structure whereas an array is a variable sized data structure.
- ii. An array is a fixed sized data structure whereas an ArrayList can change size.
- iii. Arrays are not built into the language but ArrayLists are.
- iv. None of the above

(h) Which one of the following keywords is used to access member of class with no need of object creation? [2]

- i. static
- ii. public
- iii. private
- iv. protected

(i) Which are true about the following function prototype? Select all that apply. [2]

```
static int myFunction(int x, int y);
```

- i. It does return an integer value.
- ii. It takes two parameters, both of which are int type.
- iii. It adds x and y together.
- iv. It uses pass by reference to access the variables.

(j) Given the following function:

```
1 public static String foo(LinkedList<String> list) {
2
3     Iterator<String> i = list.iterator();
4     String result = i.next();
5     while (i.hasNext()) {
6         String next = i.next();
7         if (next.length() > result.length()) {
8             result = next;
9         }
10    }
11    return result;
12 }
```

What would be printed if the following code snippet is executed:

[2]

```
LinkedList <String> list = new LinkedList<String>();
list.add("22");
list.add("Goldsmiths");
list.add("Hello");
System.out.println(foo(list));
```

- i. 22
- ii. Goldsmiths
- iii. Hello
- iv. 22GoldsmithsHello

(k) In order for the following code to be correct, what must be the type of the reference variable card?

[2]

```
1 card = new Anniversary( "Jack", 20 ) ;
2 card.greeting();
3
4 card = new Holiday( "Sofia" ) ;
5 card.greeting();
6
7 card = new Birthday( "Samir", 15) ;
8 card.greeting();
```

- i. Anniversary
- ii. Holiday
- iii. Birthday
- iv. A type that is the parent class of Anniversary, Holiday and Birthday.

(l) Which of the following describes the correct advantage of polymorphism? [2]

- i. The same program logic can be used with objects of several related types
- ii. Variables can be re-used in order to save memory
- iii. Constructing new objects from old objects of a similar type saves time
- iv. Polymorphism is a dangerous aspect of inheritance and should be avoided

(m) Can an abstract parent class have non-abstract children? [2]

- i. No as an abstract parent must have only abstract children
- ii. No as an abstract parent must have no children at all.
- iii. Yes as all children of an abstract parent must be non-abstract
- iv. Yes as an abstract parent can have both abstract and non-abstract children

(n) If a derived class object is created, which constructor is called first? [2]

- i. Base class constructor
- ii. Derived class constructor
- iii. Depends on how we call the object
- iv. None of the above

(o) What is the output of the program below: [2]

```
1  static void print(){
2      System.out.println("Good");
3  }
4
5  static void print(){
6      System.out.println("Average");
7  }
8
9  public static void main(String [] args){
10     print();
11 }
```

- i. It has a compilation error
- ii. It has a run time error
- iii. It compiles, runs and prints the string 'Good'
- iv. It compiles, runs and prints the string 'Average'

(p) Given the following method:

```
1   public static void mystery(int n) {
2       if(n==0){
3           return;
4       }
5       if(n%3 == 0){
6           System.out.println(n);
7       }
8       mystery(n+1);
9   }
```

What is printed when `mystery(-5)`; is executed?

[2]

- i. 0
- ii. -3
- iii. -5
- iv. -4

(q) What is the output of the following program?

[2]

```
1   class A{
2       public A(){
3           System.out.print("A");
4       }
5   }
6   class B extends A{
7       public B(){
8           System.out.print("B");
9       }
10  }
11  class C extends B{
12      public C(){
13          System.out.println("C");
14      }
15  }
16  public class Test{
17      public static void main(String[] a){
18          new C();
19      }
20  }
```

- i. AB
- ii. BC
- iii. ABC
- iv. CBA

(r) What is the output of the following program?

[2]

```
1 class Base{
2     int value = 0;
3     Base(){
4         addValue();
5     }
6     void addValue(){
7         value += 1;
8     }
9     int getValue(){
10        return value;
11    }
12 }
13 class Derived extends Base{
14     Derived(){
15         addValue();
16     }
17     void addValue(){
18         value += 2;
19     }
20 }
21 public class Test{
22     public static void main(String[] args){
23         Base b = new Derived();
24         System.out.println(b.getValue());
25     }
26 }
```

- i. 1
- ii. 2
- iii. 3
- iv. 4

(s) What is the output of this program?

[2]

```
1 class Base
2     {
3         public int i;
4         protected int j;
5     }
6     class Derived extends Base
7     {
8         void print()
9         {
10            j = i + 1;
11            System.out.println(i + "□" + j);
12        }
13    }
14    class Inheritance
15    {
16        public static void main(String args [])
17        {
18            Derived obj = new Derived();
19            obj.print();
20        }
21    }
```

- i. 0 0
- ii. 0 1
- iii. Run time error
- iv. Compilation error

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

[2]

```
1 class Base{
2     int x =10;
3     Base(){
4         x =10;
5     }
6     int getX(){
7         return x ;
8     }
9 }
10
11 class Derived extends Base{
12     int x =20;
13 }
14
15 public class Test1{
16     public static void main(String[] args){
17         Base obj = new Derived();
18         System.out.println(obj.getX());
19     }
20 }
```

- i. Prints 10
- ii. Prints 20
- iii. Compilation error
- iv. Run-time error

Question 2 Multiple choice c++

(a) Considering the following code, which one of the following is true?

[2]

```
1
2 class Bike {
3     private:
4         int speed;
5
6     public:
7         Bike(){
8             speed = 0;
9         }
10
11         Bike (double n){
12             new Bike();
13         }
14 };
15
16 int main(){
17     Bike bike = new Bike(10);
18 }
```

- i. speed is initialised to zero in the constructor.
- ii. speed is initialised to null in the constructor.
- iii. speed is initialised to 10 in the constructor.
- iv. the variable speed can be directly accessed from any sub-class of the Car class.

- (b) Which is the correct value and explanation for the value of the colour variable inside the bike object at the end of main? [2]

```
1
2 class MotorBike{
3     private:
4         std::string colour;
5
6     public:
7         MotorBike(std::string c){
8             colour = c;
9         }
10
11         void setColour(std::string c){
12             colour = c;
13         }
14 };
15
16 void respray(Motorbike& bike){
17     bike.setColour("green");
18 }
19 int main(){
20     MotorBike bike = new MotorBike("red");
21     respray(bike);
22     std::cout << bike.colour;
23 }
```

- i. It is red because the respray function implements pass by reference, not pass by value.
- ii. It is red because the respray function implements pass by value, not pass by reference.
- iii. It is green because the respray function operates upon the bike object, setting its class member colour to a new value.
- iv. It is not possible to know what the colour would be as the bike variable is not returned from the respray function.

(c) Which one the following would result in a call to the constructor of class Bicycle? [2]

- i. `Bicycle* bike;`
- ii. `Bicycle bike;`
- iii. `Bicycle bike = new Bicycle;`
- iv. none of the above

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

```
int x;  
x = 10.5;  
std::cout << x;
```

- i. it prints the float 10.5
- ii. it prints the integer 10
- iii. runtime error
- iv. compilation error

(e) Which one is true about the following code?

[2]

```
int x = 5;
double y = 10.5 ;
x ++;
y = x + y;
```

- i. y is 16 at the end because y is rounded up when it is added to x.
- ii. y is 16.5 because adding a double to int result in a double value.
- iii. this code will not compile as it assigns a double value to an int.
- iv. none of the above

(f) Which one is true about the following code?

[2]

```
int x = 5;
double y = 10.5 ;
x ++;
x= x + y;
```

- i. 16 because the result of adding x to y is rounded down when it is assigned to x.
- ii. 16.5 because this is the most accurate result of adding 5 to 10.5, and the CPU always aims for the most accurate result.
- iii. this code will not compile as it assigns a double value to an int.
- iv. none of the above.

(g) What is the difference between a `std::vector` data structure and an array data structure? [2]

- i. A `std::vector` is a fixed sized data structure whereas an array is a variable sized data structure.
- ii. An array is a fixed sized data structure whereas a `std::vector` can change size.
- iii. Arrays are not built into the language but `std::vectors` are.
- iv. none of the above

(h) Which of the following allows child classes to access members of a parent class? Choose all that apply. [2]

- i. `const`
- ii. `public`
- iii. `private`
- iv. `protected`

(i) Which are true about the following function prototype? Select all that apply. [2]

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

- i. It does return an integer value.
- ii. It takes two parameters, both of which are `int` type.
- iii. It adds `x` and `y` together.
- iv. It uses pass by reference to access the variables.

(j) What would be printed by this program?

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

std::string foo(std::vector <std::string> vals){
    std::string res = vals[0];
    for (int i=1;i<vals.size();i++){
        if (vals[i].length() > res.length()){
            res = vals[i];
        }
    }
    return res;
}

int main(){
    std::vector<std::string> vals;
    vals.push_back("22");
    vals.push_back("Goldsmiths");
    vals.push_back("Hello");
    std::string res = foo(vals);
    std::cout << res << std::endl;
}
```

- i. 22
- ii. Goldsmiths
- iii. Hello
- iv. 22GoldsmithsHello

- (k) In order for the following code to be correct, what must be the type of the variable card? [2]

```
1   card = new Anniversary( "Jack", 20 ) ;
2   card->greeting();
3
4   card = new Holiday( "Sofia" ) ;
5   card->greeting();
6
7   card = new Birthday( "Samir", 15) ;
8   card->greeting();
```

- i. pointer to Anniversary
- ii. pointer to Holiday
- iii. pointer Birthday
- iv. Pointer to a type that is above Anniversary, Holiday and Birthday in the class hierarchy, e.g. a Parent or a Grand Parent.

- (l) Which of the following describes the correct advantage of polymorphism? [2]

- i. The same program logic can be used with objects of several related types
- ii. Variables can be re-used in order to save memory
- iii. Constructing new objects from old objects of a similar type saves time
- iv. Polymorphism is a dangerous aspect of inheritance and should be avoided

- (m) What is the purpose of virtual functions? [2]

- i. They enable polymorphism through runtime type choice.
- ii. They enable inheritance by allowing child classes to change the implementation of the function.
- iii. They make it possible to use pointers to manipulate objects.
- iv. They make it possible to use pass by reference without using pointers.

- (n) If a child class object is created, which constructor is called first? [2]

- i. Parent class constructor
- ii. Child class constructor
- iii. Depends on how we call the object
- iv. None of the above

(o) What is the output of the program below:

[2]

```
1 #include <iostream>
2
3 void print(){
4     std::cout << "Average";
5 }
6
7 class Test{
8     void print(){
9         std::cout << "Good";
10    }
11 };
12 int main(){
13     Test t;
14     print();
15 }
```

- i. it has a compilation error
- ii. it has a run time error
- iii. it compiles, runs and prints the string 'Good'
- iv. it compiles, runs and prints the string 'Average'

(p) Given the following method:

[2]

```
1 void mystery(int n) {
2     if(n==0){
3         return;
4     }
5     if(n%3 == 0){
6         std::cout << n << std::endl;;
7     }
8     mystery(n+1);
9 }
```

What is printed when `mystery(-5);` is executed?

[2]

- i. 0
- ii. -3
- iii. -5
- iv. -4

(q) What is the output of the following program?

[2]

```
1 #include <iostream>
2
3 class A{
4     public:
5     A(){
6         std::cout << "A";
7     }
8 };
9 class B : public A{
10    public:
11    B(){
12        std::cout << "B";
13    }
14 };
15 class C : public B{
16    public:
17    C(){
18        std::cout << "C";
19    }
20 };
21 int main(){
22     C c;
23 }
```

- i. AB
- ii. BC
- iii. ABC
- iv. CBA

(r) What is the output of the following program?

[2]

```
1 #include <iostream>
2 class Base{
3 private:
4     int value;
5 public:
6     Base(){
7         value = 0;
8         addValue();
9     }
10    void addValue(){
11        value += 1;
12    }
13    int getValue(){
14        return value;
15    }
16 };
17 class Derived : public Base{
18 public:
19     Derived(){
20         addValue();
21     }
22     void addValue(){
23         value += 2;
24     }
25 };
26
27 int main(){
28     Base* b = new Derived();
29     std::cout << b->getValue();
30 }
```

- i. 1 as the constructor of class Base calls addValue on class derived.
- ii. 2 as addValue adds to to value, which starts at 0.
- iii. 3 as addValue gets called twice.
- iv. none of the above as value is private and therefore cannot be accessed from class Derived.

(s) What is the output of this program?

[2]

```
1 #include <iostream>
2 class Base
3 {
4 public:
5     int i;
6 protected:
7     int j;
8 };
9 class Derived : public Base
10 {
11 public:
12     void print()
13     {
14         j = i + 1;
15         std::cout << i << "-" << j << std::endl;
16     }
17 };
18
19 int main()
20 {
21     Derived* obj = new Derived();
22     obj->print();
23 }
```

- i. 0-0
- ii. 0-1
- iii. run time error
- iv. compilation error

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

[2]

```
1 #include <iostream>
2 class Base{
3 private:
4     int x =10;
5 public:
6     Base(){
7         x =10;
8     }
9     int getX(){
10        return x ;
11    }
12 };
13
14 class Derived : public Base{
15 private:
16     int x =20;
17 };
18
19 int main(){
20     Base* obj = new Derived();
21     std::cout << obj->getX() << std::endl;
22 }
```

- i. prints 10
- ii. prints 20
- iii. compilation error
- iv. run-time error

Part B
Written answer questions JAVA

Question 3 Data Structure – JAVA

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

```
1
2 import java.util.*;
3 public class MyClass{
4     public static void main(String [] args) {
5         String [] myArray = {"c", "b", "a"};
6         List<String> list = Arrays.<String>asList(
7             myArray);
8         TreeSet<String> set = new TreeSet<String>(
9             list);
10        set.add("d");
11        set.add("a");
12        set.remove("a");
13        String result = "";
14        Iterator<String> myIterator = set.iterator();
15        while (myIterator.hasNext())
16            result += myIterator.next();
17        System.out.println("result:␣" + result);
18    }
19 }
```

[8]

(b) Given the following method:

```
1 public static Map<String, Integer> mystery(String [] list){
2     Map<String,Integer> map = new TreeMap<String,Integer>();
3     for(int i=0; i<list.length;i++)
4     {
5         String str = list[i].toLowerCase();
6         if(map.containsKey(str))
7         {
8             map.put(str,map.get(str)+1);
9         }
10        else map.put(str,1);
11    }
12    }
13    return map;
14 }
```

i. What does the method 'mystery' do ?

[5]

ii. What will be printed if the following code snippet is executed?

[5]

```
1 String [] arr ={"the", "goldsmiths", "college", "Goldsmiths
   □college", "Goldsmiths", "College", "Computing", "
   computer", "all"};
2 System.out.println(mystery(arr));
```

(c) Write a method, *mostFrequent* (*ArrayList* < *String* > *list*), that prints the most frequent word in the list as well as the number of its occurrences

[12]

Question 4 Inheritance & Polymorphism

- (a) For each of the following, identify the line number where it is defined in the following abstract class definition.

```
1  abstract class Student {
2      public Student() { }
3      public abstract double computeFinalMark();
4      private double [] marks;
5      private double passMark=50;
6      protected String name;
7
8  }
```

- i. Constructor. [1]
- ii. Instance variable member. [2]
- iii. Class function member. [1]
- (b) Define what is mean by an 'anonymous' class. Write the code to create an instance of the Student class as an anonymous class. [6]
- (c) Write a class called **UnderGraduateStudent** that extends the Students class. The passMark for undergraduate students is only 40. For each line of your code, explain what is going on (eg. Line 1: declare a variable of type int called x). . [6]
- (d) Write a variable declaration for a variable called students that is of type ArrayList and that can store a set of students objects. [2]
- (e) Could the ArrayList be used with **UnderGraduateStudent** objects as in the code below? Explain the programming technique that is going on here and why you answered yes or no. [3]
- ```
students.add(new UnderGraduateStudent());
```
- (f) State two variable declarations that could be associated with an object of type **UnderGraduateStudent**. [2]
- (g) Define nested classes, with an example that uses this class to class relationship as opposed to using inheritance. [2]
- (h) You are designing the programmatic structure of a virtual reality simulation of classroom. State the names and purposes of THREE classes you would define in this simulation, where some have a nested relationship between them and some have an inheritance relationship. State what the relationship is between the classes. [5]

**Question 5**      Methods & Exceptions

- (a) Explain what is a checked exception and give an example of this type of exceptions. [4]
- (b) Explain why should a caller to function do parameter checking. [6]
- (c) Explain `IllegalArgumentException` and give an example to illustrate this exception.. [4]
- (d) Define a method called **Similarity** that returns the number of similarities between 2 strings. We define the similarity of two strings `str1` and `str2` as the number of indices `i` for which `str1.charAt(i) = str2.charAt(i)` [8]
- (e) Write a method called **getLeastSimilar** that takes an `ArrayList` of `Strings` and a single string as arguments, and uses the similarity function defined above to find the string in the array of strings that is least similar to the single string, and return it. [8]

## Part C

Written answer questions c++

**Question 6** Data Structure – c++

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

```
1 #include <iostream>
2 #include <map>
3 #include <string>
4 #include <vector>
5
6 int main(){
7 std::string myArray[] = {"c", "b", "a"};
8 std::map <std::string, std::string> myMap;
9 myMap[myArray[0]] = myArray[1];
10 myMap[myArray[1]] = myArray[0];
11 myMap[myArray[2]] = myArray[0];
12 myMap.erase("b");
13 std::string result = "";
14 for (int i=0;i<3;i++){
15 result += myMap[myArray[i]];
16 }
17 std::cout << "result:_" << result << std::endl;
18 }
```

[6]

- (b) Give an implementation of the missing methods from the class StringVectorWrapper, shown below.

```
1 class StringVectorWrapper {
2 public:
3 std::string getStringAt(int index){
4 // return the string at index
5 }
6 int getSize(){
7 // return the current size
8 }
9 void Display(){
10 // iterate the vector and print the strings
11 }
12 void add(std::string s){
13 // add the string s at index in the end of the vector
14 }
15 void removeAt(int index){
16 // remove the string at index from the vector
17 }
18 private:
19 std::vector <std::string> list;
20 };
```

- i. Implement getStringAt [2]
- ii. Implement getSize [2]
- iii. Implement Display [2]
- iv. Implement add [3]
- v. Implement removeAt [3]



- (c) Write a method, *mostFrequent* (*std::vector* < *std::string* > *list*), that prints the most frequent word in the list as well as the number of its occurrences.

[12]

### Question 7 Inheritance & Polymorphism

- (a) For each of the following, identify the line number where it is defined in the following class definition.

```
1 class Student {
2 public:
3 Student() { }
4 virtual double computeFinalMark();
5 private:
6 double marks[100];
7 double passMark;
8 protected:
9 std::string name;
10 };
```

- i. Constructor. [1]
- ii. Instance variable member. [2]
- iii. Class function member. [1]
- (b) Would you expect to find this code in a .h or a .cpp file? Explain why. [2]
- (c) Write the implementation for the constructor. It should initialise the passMark to 50. [4]
- (d) Write an UnderGraduate class prototype and the necessary implementation to extend the Student class. The passMark for undergraduate students should be initialised to 40. For each line of your code, explain what is going on (eg. Line 1: declare a variable of type int called x). . [6]
- (e) Write a variable declaration for a variable called students that is of type std::vector and that can store pointers to Student objects. [2]
- (f) Could the vector be used with UndergraduateStudent objects as in the code below? Explain the programming technique that is going on here and why you answered yes or no. [3]
- ```
students.add(new UnderGraduateStudent());
```
- (g) State two variable declarations for variables that could be associated with an object of type UnderGraduateStudent. [2]
- (h) Define compounding, with an example that uses this class to class relationship as opposed to using inheritance. [2]
- (i) You are designing the programmatic structure of a virtual reality simulation of classroom. State the names and purposes of THREE programmatic classes you

would define in this simulation, where some have a compounded relationship between them and some have an inheritance relationship. State what the relationship is between the classes.

[5]

Question 8 methods & Exceptions

- (a) Explain what is checked exception and give an example of this type of exceptions. [4]
- (b) Explain why the caller of a function should do parameter checking, as opposed to the function itself. [6]
- (c) Explain the difference between exception handling and value checking. [4]
- (d) Given two strings, s1 and s2, write a method **substringMax(std::string s1, std::string s2)** that finds the length of the longest contiguous string s3 such that both s1 and s2 contain s3. You assume that input strings are in lowercase. [8]

- (e) Write a method called **getMaxSubtring** that takes an vector of strings and a single string as arguments, and uses the **substringMax** function defined above to find the string in the vector of strings that contains the longest substring match to the single string, and return it. [8]