

**UNIVERSITY OF LONDON**

**GOLDSMITHS COLLEGE**

**Department of Computing**

**B. Sc. Examination 2015**

**IS52028A**

**Principles and Applications of Programming**

**Duration: 3 hours**

**Date and time:**

---

*This paper is in Three parts: part A part B and part C. There are a total of two questions in each part. **You should attempt a total of 4 questions. You should answer ALL questions in Part A and All Questions in either Part B OR Part C.***

*Full marks will be awarded for complete answers to a total of FOUR questions. Each question carries 25 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

### Question 1

(a) i. What is the difference between the following two notions: an object in an object oriented language and a variable in a structured programming language? [2]

ii. Which of the following is/are advantage of using object oriented programming?

1. Code Reusability
2. Can create more than one instance of a class without interference
3. Platform independent
4. All of the above

[3]

iii. A sub-class is placed in the same package as its super-class. You want to allow the sub-class to access a method defined in the super-class. Which of the following is/are the correct access specifier(s) for this purpose?

1. public
2. protected
3. default
4. All of these

[3]

(b) i. What will be the output of the following program?

```
class A
{
    public static void main (String[] args)
    {
        int x=0;
        System.out.println(x++);
        System.out.println(++x);
        System.out.println(--x);
        System.out.println(x--);
        System.out.println(x);
    }
}
```

[4]

ii. What will be the output of the following code fragment?

```
System.out.println((int)1.5);
```

[2]

(c) Consider the following program:

```
public class Q1c
{
    private boolean b;
    private int n;

    public Q1c(){

    }

    public boolean getB()
    {
        return b;
    }

    public int getN()
    {
        return n;
    }

    public static void main(String[] args)
    {
        Q1c q1c = new Q1c();
        System.out.println(q1c.getN());
        System.out.println(q1c.getB());
    }
}
```

Which of the following best describes what will happen?

i. It will compile with no errors and output:

0  
false

- ii. There will be a compilation error because the instance variables are not initialised
- iii. The program will compile correctly but have a run time error because the instance variables have not been initialised

[3]

(d) Consider the following two classes, A and B:

```
public class A
{
    private int n = 51120;

    public int getN()
    {
        return n;
    }

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

```
public class B
{
    private int n = 51120;

    public B (int number)
    {
        n = number;
    }

    public int getN()
    {
        return n;
    }

    public static void main(String[] args)
    {
        B b = new B();
        System.out.println(b.getN());
    }
}
```

i. One class will compile correctly and one will not. Identify which one of A and B produces a compilation error?

ii. Explain why one class compiles and why the other does not.

[4]

(e) Using the method `Math.random()`, write a complete Java program that generates a million random real numbers between 0 and 1 and prints out the largest.

[4]

## Question 2

(a) i. The concept of multiple inheritance is implemented in Java by

1. extending more than one class.
2. extending one class and implementing one or more interfaces
3. implementing two or more interfaces
4. Both 2 and 3

[3]

ii. For each of the following statements say whether it is true or false:

1. An abstract class can be instantiated
2. An abstract class cannot be instantiated
3. A class can implement any number of interfaces, but can only extend one abstract class
4. An abstract class can have concrete (ie non-abstract) methods

[3]

(b) Why will the class Q2b shown below give a compilation error?"

```
abstract class Dog
{
    public abstract String bark();
}

class Q2b
{
    public static void main(String[] args)
    {
        Dog d=new Dog();
    }
}
```

[3]

(c) Consider the following code:

```
public interface Q2c
{
    public boolean isEmpty();
    public void push(Object item);
    public Object top();
    public void pop();

    public String toString(int n, int[] a)
    {
        String s = "";
        for (int i=0; i<n; i++) s += a[i] + " ";
        return s;
    }
}
```

Which one of the following will happen?

- i. the Q2c interface produces a compilation error because the toString() method is not abstract
- ii. the Q2c interface will compile correctly
- iii. the Q2c interface will compile correctly but there will be a run-time error in any implementing classes because of the non-abstract toString() method

[5]

(d) i. Consider the following definition:

```
interface Animal
{
    abstract int numberOfFeet();
}
```

Define a (non-abstract) class Cat that implements Animal.

[5]

- ii. Suppose we have pet cats which are like cats but they also have a name.
  1. Write a class for pet cats (include a constructor) and a method for getting the cat's name.
  2. Write a method which has an ArrayList a of pet cats as its single parameter, and returns the pet cat in a with the longest name.

[6]

## Part B

### Question 3

(a) Consider the following code:

```
import java.util.*;
public class LinkedListDOSomething{
    public static void main(String[] args)
    {
        LinkedList <String> words = new LinkedList<String>();
        words.add("hello");
        words.add("goodbye");
        words.add("this");
        words.add("that");
        words.add("Good morning");
        DoSomethingtoWords(words);
        for (int i=0; i < words.size(); i++) {
            System.out.println(words.get(i));
        }
    }
    public static void DoSomethingtoWords (LinkedList<String> list)
    {
        Iterator<String> i = list.iterator();
        while (i.hasNext()) {
            String element = i.next();
            if (element.length() % 2 == 0) {
                i.remove();
            }
        }
    }
    public static void DoSomethingElsetoWords (LinkedList<String> list){
        Iterator<String> i = list.iterator();
        while (i.hasNext()) {
            String element = i.next();
            if (element.length() % 2 == 1) {
                i.remove();
            }
        }
    }
}
```



- i. Explain the meaning of the line `import java.util.*;` [5]
  - ii. Explain the meaning of `Iterator <String> i = list.iterator();` [5]
  - iii. What is observed when `main()` method is executed? [5]
- (b) The Sieve of Eratosthenes is an ancient algorithm that generates prime numbers. Consider the list of numbers 2,3,4, 5,6,7,8,9,10. The algorithm starts with the first prime in the list, which is 2, and then iterates through the remainder of the list removing any number that is a multiple of 2 (in this case) leaving 2,3,5,7,9. We then repeat with the second prime number in the list (which is 3) and then iterate through the list removing all multiples of 3, leaving (in this case) 2,3,5,7. This procedure is repeated with each successive prime number leaving a list of only primes. Implement this algorithm using an `ArrayList` of integers that is initialised from 2 to 100. You must use an iterator to scan through the data.

[10]

#### Question 4

(a) Briefly describe Object Serialisation and why it is useful. [8]

(b) The following represents a snippet of code that inputs two integers and divides them:

```
Scanner scan = new Scanner(System.in);

int n1, n2;

double r;

n1=scan.nextInt();

n2=scan.nextInt();

r=(double)n1/n2;
```

Place this code into a try-catch block with multiple catches so that different error messages are printed to the screen if we attempt to divide by zero or textual data is input. If any of these errors occur the program should loop back and allow the user to enter the data again. [7]

(c) In graphics programming what is a layout policy in awt and Swing? Give code snippets to illustrate your answer. [10]

## Part C

### Question 5

(a) Consider the following code:

```
#include <iostream>
using namespace std;
class IntStack{
    private:
        int *stackArray;
        int stackSize;
        int top;
    public:
        IntStack(int);
        void push(int);
        void pop(int &);
        bool isFull(void);
        bool isEmpty(void);
};

IntStack::IntStack(int size){
    stackArray = new int[size];
    stackSize = size;
    top = -1;
}

void IntStack::push(int num){}
void IntStack::pop(int &num){}
bool IntStack::isFull(void){}
bool IntStack::isEmpty(void){}
int main(){
    IntStack stack(5);
    int catchVar=9;
    stack.push(3);
    stack.push(5);
    stack.push(5);
    stack.push(3);
    stack.push(25);
    stack.pop(catchVar);
    stack.pop(catchVar);
    stack.pop(catchVar);
    cout << catchVar << endl;
    stack.pop(catchVar);
    cout << catchVar << endl;
}
```

```
    stack.pop(catchVar);  
    cout<<catchVar<<endl;  
    return 0;  
}
```

- i. Explain the purpose of the `isFull()`, `pop()` and `push()` functions and how they could be implemented (you do not have to code them). The `isFull()` function checks whether the stack is full by returning an appropriate Boolean value. `Pop` and `push` have their usual meanings. You can include pseudo code in your answers. [10]
- ii. What is the output for the code above? (You should assume that the functions `isFull()`, `isEmpty()`, `pop()` and `push()` are all implemented correctly). [5]
- (b) The Sieve of Eratosthenes is an ancient algorithm that generates prime numbers. Consider the list of numbers 2,3,4, 5,6,7,8,9,10. The algorithm starts with the first prime in the list, which is 2, and then iterates through the remainder of the list removing any number that is a multiple of 2 (in this case) leaving 2,3,5,7,9. We then repeat with the second prime number in the list (which is 3) and then iterate through the list removing all multiples of 3, leaving (in this case) 2,3,5,7. This procedure is repeated with each successive prime number leaving a list of only primes. Implement this algorithm using a `std::vector` of integers that is initialised from 2 to 100. You must use an iterator to scan through the data. [10]

## Question 6

(a) Consider the following code:

```
#include<iostream>
using namespace std;
class A {
    public:
    void f() {
        cout << "A::f" << endl;
    }
    virtual void g() {
        cout << "A::g" << endl;
    }
};

class B : public A {
    public:
    void f() {
        cout << "B::f" << endl;
    }
    virtual void g() {
        cout << "B::g" << endl;
    }
};

int main() {
    A a;
    B b;
    A* aPtr = &a;
    A* bPtr = &b;
    aPtr->f();
    aPtr->g();
    bPtr->f();
    bPtr->g();
    return 0;
}
```

What is the output of the code above? Explain your answer in terms of compile-time polymorphism.

[5]

(b) C++ allows for the creation of Template functions. What is meant by a Template functions in C++ and how (and why) would a programmer create a specialisation of a Template function. Give snippets of C++ code to illustrate your comprehension.

[15]

(c) Describe the use of C++ iterators in the Standard Template Library, demonstrate your comprehension by including snippets of code.

[5]