

Question 1

Given Program B1 as shown below. Answer questions (a) and (b) following it.

```
1 //Program B1
2 class classA {
3     private String var1;
4     private classB[] classBList;
5     private int numClassB;
6
7     public classA(String v) {
8         var1 = v;
9         classBList = new classB[5]; }
10
11     public void addMemberClassB(classB cB) {
12         classBList[numClassB] = cB;
13         numClassB++; }
14
15     public void print() {
16         System.out.println("\nClass B: " + var1);
17         System.out.println("Number of object(s) in Class B: " +
18             numClassB);
19         System.out.println("List of object(s) in Class B");
20         for (int i = 0; i < numClassB; i++)
21             System.out.println(">> " + (i+1) + " << " +
22                 classBList[i].getVar2());
23         System.out.println(); }
24 }
25
26 class classB {
27     private int var2;
28     private classC cC;
29
30     public classB(int v) {
31         var2 = v; }
32
33     public classB(classC c) {
34         cC = c; }
35
36     public int getVar2() {
37         return var2; }
38
39     public String toString() {
40         return cC + "\n"; }
41 }
42
43 class classC
44 {
45     private String var3;
46
47     public classC(String v) {
48         var3 = v; }
49
50     public String toString() {
51         return "Hello " + var3 + "! I'm Class C"; }
52 }
53
54 public class ClassRelationApp {
55     public static void main(String [] args) {
56
57         _____
58         _____
59
60         for (int i = 0; i < 2; i++)
61             cA.addMemberClassB(cB1[i]);
62
63         cA.print();
```

64	
65	
66	System.out.println(cC); }
67	}

- Draw the UML class diagram that shows the relationship among classes in Program B1.
- Based on the output in Figure 1, write codes to create the object(s) for appropriate class with appropriate argument(s) to complete the **main()** method in **ClassRelationApp** class. Write your answer in Table 1.

```

Class B: Hi! I'm Class A
Number of object(s) in Class B: 2
List of object(s) in Class B
>> 1 << 221100
>> 2 << 121212

Hello Liana! I'm Class C

```

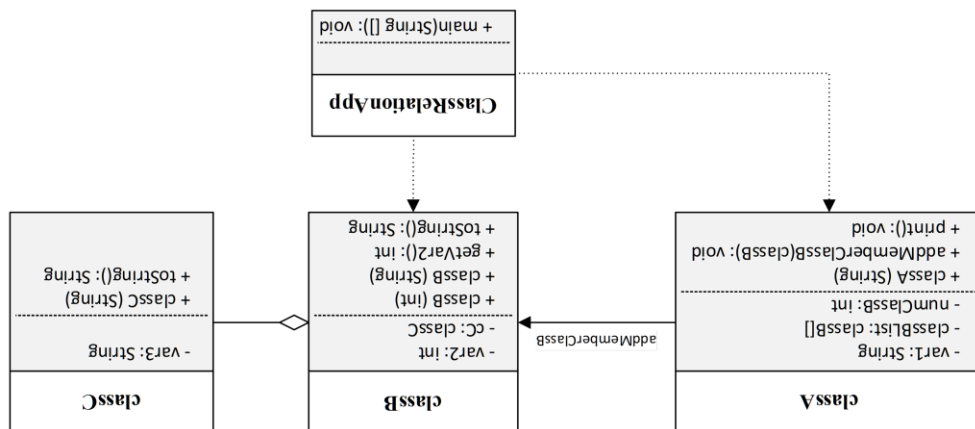
Figure 1: Screen output

Table 1: Answer for Question 1

Line	Java Statement(s)
56	
57	
64	
65	

Line	Java Statement(s)
77	classB [] cB1 = {new classB(221100), new classB(121212)};
78	classA cA = new classA("Hi! I'm Class A");
85	classC cC = new classC("Liana");
86	classB cB2 = new classB(cC);

(b)



(a)

Answer:

Question 2

Given the following three classes: **Badminton**, **Football**, and **Sport**, answer questions (a) and (b).

```
1 public class Badminton {
2     public void m1() {
3         System.out.println("Racket");
4     }
5     public void m2() {
6         System.out.println("Shuttle");
7     }
8     public String toString() {
9         return ("Ole");
10    }
11 }
```

```
1 public class Football extends Badminton {
2     public void m1() {
3         System.out.println("Boots");
4     }
5     public void m2() {
6         super.m1();
7     }
8     public String toString() {
9         return super.toString() + super.toString();
10    }
11 }
```

```
1 public class Sport {
2     public static void main(String[] args) {
3         Football mycourt = new Football();
4         System.out.println(mycourt);
5         mycourt.m1();
6         mycourt.m2();
7     }
8 }
```

- i. What is the expected output when the above program is executed?
- ii. Consider the new codes in the class **Sport** and the output as in Figure 2:

```
1 public class Sport {
2     public static void main(String[] args) {
3         Hockey stick = new Hockey();
4         System.out.println(stick);
5     }
6 }
```

```
Racket
Boots
Hockey game!
```

Figure 2: Output

Write a **complete** class **Hockey** in Figure 3 and **add** a **NEW** constructor in class **Football** in Figure 4, which contains **super** statement in order to produce the output on Figure 2.

```
1 class Hockey _____ {
2
3
4
5
6
7
8
9
```

Figure 3

```
1 public class Football extends Badminton {
2     _____
3     _____
4     _____
5
6     public void m1() {
7         System.out.println("Boots");
8     }
9     public void m2() {
10        super.m1();
11    }
12    public String toString() {
13        return super.toString() + super.toString();
14    }
15 }
```

Figure 4

Answer:
(i) OleOle
Boots
Racket

```
(ii)
class Hockey extends Football {
    Hockey() {
        super.m1();
    }
    public String toString() {
        return "Hockey game!";
    }
}

public class Football extends Badminton {
    Football() {
        super.m1();
    }
    public void m1() {
        System.out.println("Boots");
    }
    public void m2() {
        super.m1();
    }
    public String toString() {
        return super.toString() + super.toString();
    }
}
```

Question 3

Complete the class declaration of **Calculator**, **Addition**, **Multiplication**, and interface class **Operation** based on the comments given in Program B2. Class **Addition** and class **Multiplication** will inherit class **Calculator** and implements class **Operation**.

```
1 // Program B2
2 // i.declare interface class Operation
3 _____ {
4 public int sum();
5 }
6 //ii. declare class Calculator
7 _____ {
8 protected int no1;
9 protected int no2;
10 //iii. declare 2 arguments constructor class Calculator
11 _____ {
12     _____
13     _____
14 } }
15 // iv.define class Addition that inherit class Calculator and
16 // implements interface class Operation
17 _____ {
18 // v.declare constructor of Addition that inherit attributes of
19 // class Calculator
20 _____{
21     _____
22 }
23
24 //vi.define abstract method sum
25 _____{
26     return no1 + no2;
27 } }
28
29 //vii.define class Multiplication that inherit class Calculator and
30 //implements interface class Operation
31 _____{
32 // viii.declare constructor of Multiplication that inherit attributes
33 // of class Calculator
34 _____{
35     _____
36 }
37 //ix.define abstract method sum
38 _____ {
39     return no1 * no2;
40 }
41 }
42
43 class InterfaceClass {
44     public static void main(String args[]){
45 //x.Create object of Addition with argument 5 and 8
46     _____
47
48 //xi.Create object of Multiplication with argument 4 and 3
49 //
50
51     System.out.println ("Sum of Addition: " + a.sum());
52     System.out.println ("Sum of Multiplication: " + m.sum());
53 }
54 }
55
```

Answer:

```
1 // Program B2
2 // declare interface class Operation
3 interface Operation {
4     public int sum();
5 }
6 //declare class Calculator
7 class Calculator {
8     protected int n1;
9     protected int n2;
10    // declare constructor class Calculator
11    Calculator2 (int nom1, int nom2) {
12        nom1=nom1;
13        nom2=nom2;
14    }
15    // define class Addition that inherit class Calculator and
16    // implements interface class Operation
17    class Addition extends Calculator2 implements Operation {
18        // declare constructor of Addition that inherit attributes of
19        // class Calculator
20        Addition (int n1, int n2) {
21            super(n1,n2);
22        }
23        //define abstract method sum
24        public int sum() {
25            return n1 + n2;
26        }
27        //define class Multiplication that inherit class Calculator and //implements
28        interface class Operation
29        class Multiplication extends Calculator2 implements Operation{
30            // declare constructor of Multiplication that inherit attributes
31            // of class Calculator
32            Multiplication (int n1, int n2) {
33                super(n1,n2);
34            //define abstract method sum
35            public int sum() {
36                return n1 * n2;
37            }
38        }
39        class InterfaceClass {
40            public static void main(String args[]) {
41                //Create object of Addition with argument 5 and 8
42                Addition a = new Addition (5, 8);
43                //Create object of Multiplication with argument 4 and 3
44                Multiplication m = new Multiplication (4, 3);
45                System.out.println ("Sum of Addition: " + a.sum ()
46                )
47                System.out.println ("Sum of Multiplication: " + m.sum
48                )
49            }
50        }
```

Question 4

- a. Consider the given Program B3 below. Answer question (i) to (v).

```
1 // Program B3
2 import java.util.Scanner;
3
4 class Example2 {
5     public static void main(String[] args) {
6
7         int a, b, result;
8
9         Scanner input = new Scanner(System.in);
10        System.out.println("Please enter two integer numbers :");
11
12        a = input.nextInt();
13        b = input.nextInt();
14
15        result = a / b;
16
17        System.out.println("Result = " + result);
18    }
19 }
```

- i. Write the output of the program, if the following inputs are entered. If error, specify the type of the error.

Please enter two integer numbers :

6 2

Please enter two integer numbers :

5 0

- ii. Now modify the program to handles exceptions.

- b. Complete class **Example2** by answering the question (i) to (ix) based on the comments given in Program B4.

```
1 // Program B4
2 import java.util.Scanner;
3
4 public class Example2 {
5     public static void main(String args[]) {
6         try {
7             Scanner input = new Scanner(System.in);
8             String _____(i)_____; // declare a string variable
9
10            _____(ii)_____; // declare another related variable
11            System.out.println ("Enter an integer:");
12            inputData = input.nextLine();
13            num = Integer.parseInt(inputData);
14            System.out.println ("The square is: "+(num*num));
15            method();
16        }
```

17	
18	_____ (iii) _____ (____ (iv) _____ ex) {
19	System.out.println ("Wrong data type");
20	}
21	
22	_____ (v) _____ (____ (vi) _____ ex) {
23	System.out.println ("Array index out of bound");
24	}
25	
26	_____ (vii) _____ (Exception ex) {
27	System.out.println ("Input problem");
28	}
29	}
30	
31	_____ (viii) _____ void method() _____ (ix) _____ Exception {
32	String languages[] = { "C++", "Java", "Perl"};
33	
34	for (int i = 1; i <= 3; i++) {
35	System.out.println(languages[i]);
36	}
37	}
38	}

Table 2a

Table 2b

i	
ii	
iii	
iv	
v	

vi	
vii	
viii	
ix	

iii. What is the expected output if the input entered is 4?

Answer:

a(i) Result = 3

Error message of ArithmeticException

a(ii)

import java.util.Scanner;

class Example2 {

public static void main(String[] args) {

int a, b, result;

Scanner input = new Scanner(System.in);

System.out.println("Input two integers");

a = input.nextInt();

b = input.nextInt();

try {

result = a / b;

System.out.println("Result = " + result);

}

catch (ArithmeticException e) {

System.out.println("Exception caught: Division by zero.");

} }

b(i)

i	inputData
ii	int num
iii	catch
iv	(NumberFormatException)
v	catch

vi	(ArrayIndexOutOfBoundsException)
vii	catch
viii	static
ix	throws

Output:

Enter an integer:

4

The square is: 16

Java

Perl

Array index out of bound