**Group members:**

Xinyu Qu (ID:1469135)

Mingjun Wu (ID:1490523)

**Question 1: (**HelloWorld.java)

**Question 2: (**HelloWorld\_withParam.java)

**Question 3:** (conditions.java)

x = 6 & y = 9

[output]: Have a nice day!

x = 5 & y = 2

[output]: x = 5 and y = 5

**Question 4: (**PowersOfTwo.java)

[output:]

0 1

1 2

2 4

3 8

4 16

5 32

6 64

**Question 5: (**Cubes.java)

[output:]

1 1

2 8

3 27

**Question 6: (**Cubes.java / TestCubes.java)

**Question 7:** (Vectors.java (for 7) / Matrics.java (for 7b))

[output:]

|  |  |
| --- | --- |
| i | sum |
| 0 | 0.15 |
| 1 | 0.21 |
| 2 | 0.25 |

**Question 8: (**exception.java)

class Example1 {

public static void main(String[] args) {

int temperature = 0;

if (args.length > 0) {

try {

temperature = Integer.parseInt(args[0]);

}

catch(NumberFormatException e) {

System.out.println("Must enter integer as first argument.");

return;

}

}

else {

System.out.println("Must enter temperature.");

return;

}

// Create a new coffee cup and set the temperature of

// its coffee.

CoffeeCup cup = new CoffeeCup();

cup.setTemperature(temperature);

// Create and serve a virtual customer.

VirtualPerson cust = new VirtualPerson();

VirtualCafe.serveCustomer(cust, cup);

}

}

**Explain the goal of those codes:**

* Check whether the command has an argument. If the argument is not a number, it will throw an exception and print a message “Must enter integer as first argument.”, otherwise, it will transfer the type from String to Integer.
* If the temperature is not set in argument; the app will print a message “Must enter temperature”
* Create a coffee object named cup and then using method (setTemperature) to set the temperature for the cup
* Create a VirtualPerson object named cust and serve a virtual customer through using the method named “serveCustomer”

**Question 9:** (Prime.java)

**Question 10:**

(Abst\_Animal.java/ Animal.java/ Dog.java/ Tiger.java/ InterfaceAndAbstract.java)

**The differences between Interface and abstract class:**

* **Abstract class:**

1. Abstract class can extend only one class or one abstract class at a time
2. Abstract class can extend from a class or from an abstract class
3. Abstract class can have both abstract and concrete methods
4. A class can extend only one abstract class
5. In abstract class keyword ‘abstract’ is mandatory to declare a method as an abstract
6. Abstract class can have protected, public and public abstract methods
7. Abstract class can have static, final or static final variable with any access specifies

* **Interfaces:**

1. Interface can extend any number of interfaces at a time
2. Interface can extend only from an interface
3. Interface can have only abstract methods
4. A class can implement any number of interfaces
5. In an interface keyword ‘abstract’ is optional to declare a method as an abstract
6. Interface can have only public abstract methods i.e. by default
7. Interface can have only static final (constant) variable i.e. by default