FINAL EXAM, answers

Intermediate Object-Oriented Programming (La Trobe University)

Final Examination: Semester 1, 2020, La Trobe University

Subject Code: CSE1IOO/CSE4IOO

Question 1

```
public class GradeHomework extends Homework
{
        private int weight;
        //over loaded constructor with parameter number, topic, weight
        public GradeHomework(int number, String topic, int weight)
        {
                super(number, topic);
                this.weight = weight;
        }
        //over loaded constructor with parameter number, topic
        public GradeHomework(int number, String topic)
        {
                super(number, topic);
                this.weight = 0;
       //setting the weight value to zero
        }
      //method to return the weight
        public int getNewWeight()
        {
                return weight;
        }
        //method to return the details as required
        public String getDetails()
        {
         return "Id: " + getNumber() +
         ", Name: " + getTopic() +
         ", Weight: " + weight;
        }
}
```

Question 2

```
public class Book
{

//variables declared
private String id;
private double price;


//over loaded constructor with parameter
public Book(String id, double price)throws InvalidBookPriceException
{
if(id.length() < 6)

//error message displayed if the string has less than 6 character
throw new InvalidBookPriceException("Invalid id");

else
this.id=id;

if(price < 0.0)
throw new InvalidBookPriceException("Invalid price");

else
this.price = price;


//method to set the price of the books
public void setPrice(double price)
{

//error message displayed if the string has less than 0  or less
if(price < 0.0)
throw new InvalidBookPriceException("Invalid price");


else
this.price=price;
}

}
}
```

Question 3

```
public static void main(String[] args) throws Exception
{
	Scanner infile = new Scanner(new File("persons.txt"));

	while(infile.hasNext())
	{
		String line = infile.nextLine();
		StringTokenizer tokenizer = new StringTokenizer(line, ";");
		String id = tokenizer.nextToken().trim();
		String name = tokenizer.nextToken().trim();
		String listOfHobbies = tokenizer.nextToken().trim();
		//split function used to get the number of hobbies from the string listofHobbies
		String hobbies = listOfHobbies.split(";");
		//if number of hobbies is greater than 1
		if(hobbies.length() >1)
		{
		System.out.println(name+"("+id+")has"+hobbies.length()+"hobbies");
		}
		else
		//if number of hobbies is equal to 1
		{
		System.out.ptintln(name+"("+id+")has"+hobbies.length()+"hobbies");
		}
	}

	infile.close();
}
```

Question 4

4a

```
public class ProcessFiles
{
//constructing new file named demo
   public static void main (String args[])
  {
     File folder = new File("mydir/demo");
   }
}
```

4b

```
public static void listFiles(File folder)
{
  Files check[ ] = folder.listFiles();
//printing name of file
  for(File file: check)
{
  System.out.println(file.getName());
 }
}
```

Question 5

A)

```
public static void recursiveCountUp(int low, int high)
{
//finding the lowest value
	if(low<= high)
	{
		System.out.println(""+low);
		low++;
		recursiveCountUp(low,high);
	}
}
```

B)

```
public static void countUp(int n)
{
//taking one as the lowest value as if would be the first value when we run the function
	recursiveCountUp(1,n);
}
```

Question 6

```
public static int maximum(int n1, int n2)
 { if (n1 >= n2) >=0
 {
return n1;
}
else
{
return n2;
 }
}
public static int maximum(int n1, int n2, int n3)
{
int max = maximum(n1, n2);
max = maximum(max, n3);
return max;
 }
```

Question 7

7a)

```
public class Insurance implements Commissioned

{
private String id;
private double premium;
private double commissionRate;

//overloaded constructer with attribute premium, commissionRate

public Insurance(String id, double premium, double commissionRate)

{
this.id = id;
this.premium = premium;
this.commissionRate = commissionRate;
}

//method used to return the the commission

public double getCommission()

{
return commissionRate * premium;
}


}
```

7b)

```
//method to display total commission value for all insurance

public static void insuranceCommisionTotal(ArrayList <Commissioned> commisions)
{
for (int i = 0; i< commisions.size();i++)
{
System.out.println(""+ commisions.get(i).getCommission());
}
}
```

7c)

```
//return id

public String getStringKey()
{
return id;
}
```