## Q1 (3 marks)

The following code segment is for a valid Java program. Fill each blank with the most appropriate data type available in Java.

```
___________  rentPerWeek=1000;
___________  weeksInAYear=52;
___________  rentPerYear=rentPerWeek*weeksInAYear;
___________  monthsInAYear=12;
___________  duration="month";
___________  rentPerMonth=rentPerYear/monthsInAYear;
___________  msg="Rent per "+duration+" is "+rentPerMonth;
```

## Q2 (5+2=7 marks)

Circle the locations of any syntax errors in the following Java program:

```
public Class SecondQuestion {
	public void static main(String[] args) {
		int num=13;
		int i=2;
		while (i<num /2 && ) num %%i!=0) {
			n++;
		}
		System.out.print(num + " is ");
		if (i<num / 2)
			System.out.print("not ");
		System.out.println("prime");
	}
}
```

If the syntax errors are corrected, what will be the output of the program? (Write below)

## Q3 (10 marks)

Write a Java method named *getSignAndParity* that takes an integer number as a parametric input and returns a String saying if it is positive or negative and if the number is odd or even. You must use "`else if`" and you must not have any redundancies in your checks. The following table shows some example parametric values and the corresponding return values. Your code must work with all valid values, including those not shown below.

| Number | Classification |
|---|---|
| -2147483648 | Negative even |
| -2 | Negative even |
| -1 | Negative odd |
| 0 | Positive even |
| 1 | Positive odd |
| 12434 | Positive even |

Write your code in the space below:

## Q4 (15 marks)

Write a full Java program that will take an arbitrary number of numerical values as keyboard inputs, then finally displays the values that were not repeated. At the start, you must ask the user how many values they would like to process and this input must be validated (you can assume that entered values will be of the requested data type). If the user enters a number that is outside the valid range, you must ask the user to re-enter that value until they enter a correct one, before doing anything else).

For example, if the number of values is 4 and the values are 12,34,-56 and 12, the program must output "34 was unique" and "-56 was unique". This is just an example; when written correctly, your program will work with any number of inputs.

Write your answer in the space below:

**Q5 (15 marks)**

Using good Object Oriented design practices covered in the class materials, write a class named PizzaMenu and another named Pizza as specified below.

A Pizza cannot be created without a name and a price. It also has relevant accessor methods (but no mutators). A Pizza must not have any methods that will display messages.

A PizzaMenu object cannot be created without a number of maximum pizzas. Each PizzaMenu object also has an array of Pizza objects (the size of this array can be set to the maximum number of pizzas specified) and the number of currently added pizzas. A PizzaMenu also has the methods *addPizza* and *findPizzas*. The *addPizza* method asks the user to enter a name and a price of a pizza, creates a new Pizza object and adds it to the pizza object array of that PizzaMenu object (it should also ensure that the number of added pizzas is maintained correctly and the maximum limit is not exceeded). The *findPizzas* method asks the user to enter some text, searches that text within the names of the pizzas and displays any matches. If not found, it displays "no such pizzas".

Write your code below: