| Sec | # | Student ID | Name |
|---|---|---|---|
| | | | |

# Midterm Exam

## 254275 Object Oriented Programming

Tuesday 14th January 2020
08.00-11.00 (3 hours)

**Total: 60 marks (30%)**

# DO NOT OPEN UNTIL INSTRUCTED

# ห้ามเปิดก่อนได้รับอนุญาต

**Examination Rules**

1) Do not talk in the exam room at any time. Do not make any attempt to communicate with other students.
**ห้ามพูดในห้องสอบตลอดเวลา อย่าพยายามสื่อสารกับนักเรียนคนอื่น**

2) You are not allowed to leave the exam room in the first 30 minutes and in the last 15 minutes.
**คณไม่ได้รับอนุญาตให้ออกจากห้องสอบในช่วง 30 นาทีแรก และในช่วง 15 นาทีสุดท้าย**

3) If you need to leave the exam room then raise your hand and wait for an assistant. **ถ้าต้องการไปออกจากห้องสอบให้ยกมือและรอผู้คุมสอบพาไป**

4) Read the questions carefully. If you do not understand the question then raise your hand and wait for an assistant.
**อ่านคำถามอย่างละเอียด ถ้าคุณไม่เข้าใจคำถามให้ยกมือขึ้นและรอผู้คุมสอบ**

5) Answer all questions in English only.
**ตอบคำถามทั้งหมดเป็นภาษาอังกฤษเท่านั้น**

# Part 1: Java programming

1. **Tick** ( / ) the description that matches the Java keyword or type. [6]

| | Keyword | Primitive data type | Class type | None |
|---|---|---|---|---|
| boolean | ☐ | ☑ | ☐ | ☐ |
| return | ☑ | ☐ | ☐ | ☐ |
| break | ☑ | ☐ | ☐ | ☐ |
| String | ☐ | ☐ | ☑ | ☐ |
| class | ☑ | ☐ | ☐ | ☐ |
| short | ☐ | ☑ | ☐ | ☐ |

2. This code has an **error**. Where and why? [2]

```
1   String[] names = new String[4];
2   names[0] = "Anna";    names[1] = "Bell";
3   names[2] = "Charlie"; names[3] = "Deidre";
4   for (int i = 0; i <= 4; i++) {
5       System.out.println("Hello" + names[i]);
6   }
```

Line: ....4..... Reason: ..Index 4 out of bounds for length 4.....

change to for(int i=0; i < names.length; i++) {.....

3. Write the type of these expressions. [4]

a) `5 > 10` Type: ...boolean.....

b) `2 * 1.0` Type: ...double.....

c) `"100".length()` Type: ...int.....

d) `{1, 4, 7}` Type: ...int[].....

4. Write the output when the following code is run with the given input. [3]

```
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
for (int i = 10; i > 0; i--) {
    if (i % x == 0) {
        System.out.print("Fizz ");
    } else {
        System.out.print(i + " ");
    }
}
```


a) Input: 3 Output: 10 Fizz 8 7 Fizz 5 4 Fizz 2 1

b) Input: 5 Output: Fizz 9 8 7 6 Fizz 4 3 2 1

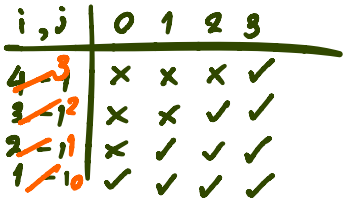
c) Input: 8 Output: 10 9 Fizz 7 6 5 4 3 2 1

5. Write the output when the following code is run with the given input. [3]

```
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
for (int i = x; i > 0; i--) {
    for (int j = 0; j < x; j++) {
        if (j >= i - 1) {
            System.out.print("*");
        } else {
            System.out.print(" ");
        }
    }//for j
    System.out.println();
}//for i
```

x = 1; 1 line, 1 item
→ ✓

x = 4; 4 line, 4 item




a) Input: 1 Output: *

b) Input: 4 Output: *

* *

* * *

* * * *

6. Complete the code by writing a function that calculates the **average** marks. [4]

Expected output:
```
3.5
```

```
class Question6 {

   public static void main (String[] args) {
         double[] marks = { 2.0, 5.0, 3.0, 4.0 };
         double average = averageMark(marks);
         System.out.println(average);
   }

   private static double averageMark(double[] marks) {
         double total = 0;
         for (double sum : marks)
               total += sum;
         return total / marks.length;
   }

}
```

7. Complete the code. [2]

Expected output:
```
5 10 15 20 25
```

```
public static void main(String[] args) {
   for (int i = 0; i < 5; i++) {
         System.out.print(5*(i+1) + " ");
   }
}
```

8. Write a program that inputs a number ***x*** and outputs the sum of *1* + *2* + … + *x*. [4]

| Input: | Output: | Input: | Output: |
|---|---|---|---|
| 5 | 15 | 3 | 6 |

```
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
```

```
int sum = 0;
for(int i = 1; i <= x; i++)
    sum += i;
System.out.print(sum);
```

9. Write a program that reads in a list of words and outputs the shortest word. [4]

| Input: | Output: | Input: | Output: |
|---|---|---|---|
| big,brown,bear | big | five,six,seven,eight | six |

```
Scanner sc = new Scanner(System.in);
String input = sc.nextLine();
```

```
String[] words = input.split(",");
String result = words[0];
for(String check : words){
    if(check.length() < result.length)
        result = check;
}
System.out.print(result);
```

## Part 2: Objects and classes

10. Describe the errors in the **Song** class. [12]

```
1   public Song {
2       private String title;
3       private String artist;
4       private duration;
5
6       public Song(String title, String artist, double duration) {
7             this.title = title;
8             this.artist = artist;
9             this.duration = duration;
10
11      public void toString() {
12            return title " - " artist;
13      }
14
15      public double getDuration() {
16            return duration;
17      }
18
19      public void setDuration(double duration) {
20            this.duration = duration;
21      }
22
23      public boolean isSameDuration(double s) {
24            return duration == s.getDuration();
25      }
26  }
```

Line: 1 Reason: public class Song {

Line: 4 Reason: private double duration;

Line: 11 Reason: public String toString() {

Line: 12 Reason: return title + " - " + artist;

Line: 23 Reason: public boolean isSameDuration(Song s) {

Line: ..... Reason: .....

11. Give the line number from the Song class (Question 10) matching the description. [4]

a) The constructor. Line: 6

b) A method with no parameters. Line: 11, 15

c) A method that does not return anything. Line: 19

d) A field (class variable) that is a Class type Line: 2, 3

Questions 12-16 require you to add methods to the Language class.
คำแปล: consonants พยัญชนะ vowels สระ letters ตัวอักษร

```
public class Language {
    private int consonants;
    private int vowels;

    // TODO answer questions 12-16
}
```

12. Write **a constructor** for the Language class [2].

```
public Language(int consonants, int vowels) {
    this.consonants = consonants;
    this.vowels = vowels;
}
```

13. Write methods for **getConsonants** and **getVowels**. [3]

```
public int getConsonants() {
    return consonants;
}
public int getVowels() {
    return vowels;
}
```

Examples:

```
Language english = new Language(21, 5);
Language arabic = new Language(29, 0);

System.out.println(english.getLetters());
// Output: "26"

System.out.println(english.toString());
// Output: "21 consonants, 5 vowels, total 26 letters"

System.out.println(arabic.hasNoVowels());
// Output: true
```

14. Write a method that returns the **sum** of consonants and vowels called **getLetters**. [2]

```
public int getLetters() {
    return consonants + vowels;
}
```

15. Write a **toString** method (following example output above). [3]

```
public String toString() {
    return consonants + " consonants, " + vowels +
        " total " + getLetters + " letters";
}
```

16. Write a method **hasNoVowels** that **returns true** if the language has 0 vowels. [2]

```
public boolean hasNoVowels(Language l) {
    return (l.getVowels() == 0);
}
```

- END -