

## CSEN202: Introduction to Computer Science Spring Semester 2017

### Midterm Exam

#### Bar Code

**Instructions: Read carefully before proceeding.**

- 1) Please tick your major

	<b>Major</b>
	Civil
	BI
	Engineering

- 2) Duration of the exam: 2 hours (120 minutes).
- 3) No books or other aids are permitted for this test.
- 4) This exam booklet contains 11 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete.**
- 5) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the three extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets.**
- 6) When you are told that time is up, stop working on the test.

**Good Luck!**

---

Don't write anything below ; -)

Exercise	1	2	3	4	5	$\Sigma$
Possible Marks	16	10	12	12	15	65
Final Marks						

**Exercise 1**

(16 Marks)

For each code fragment, (1) fill in the most appropriate data type in the 1st column and (2) fill in the value that z contains after the code has been executed in the 2nd column. If the code would result in an error, write **ERROR** in the 1st column and give the reason for the error in the 2nd column (you do not need to write the exact error message). The first two have been done for you.

Code	Data Type/Error	Value of z/Reason
<code>__ z = 0.0; z++;</code>	double	1.0
<code>__ z = true; z = 1;</code>	ERROR	z appears to have been declared as a boolean originally and so can't be set to an int.
<code>__ z = 8; z /= 3;</code>	int	2
<code>String s = "GUC"; __ z = s.length();</code>	int	3
<code>__ z = 'a'; z++;</code>	int or char	98 or b
<code>__ z = "hello"; z = z + 11.0;</code>	String	hello11.0
<code>String s = "CSEN"; __ z == "CSE" + "N";</code>	ERROR	== used for equality check not for assignment
<code>String s = "apples"; __ z = s.charAt(5);</code>	char	s
<code>int x = 0; int y = 1; __ z = (x = y);</code>	int	1
<code>__ z = 'a' &lt; 'b';</code>	boolean	true

**Exercise 2**

(8+2=10 Marks)

- a) Write a method that takes two strings `s1` and `s2` as inputs and checks whether string `s1` is a substring in string `s2`.

You are only allowed to use the methods `charAt`, `length`, and `equals`. Do not use neither `indexOf` nor `contains` method.

```
checkOccurrences("ab", "ababcdab") -> true
checkOccurrences("ababcdab", "ab") -> false
checkOccurrences("aa", "aaaaa") -> true
```

**Solution:**

```
public static boolean checkOccurrences (String s1, String s2)
{
    boolean f = false;
    for (int i = 0; i < s2.length(); i++)
    {
        String temp = "";
        for(int j = 0; j < s1.length() && (i+s1.length()) <= s2.length(); j++)
        {
            temp += s2.charAt(i+j);
        }
        if(temp.equals(s1))
            f = true;
    }
    return f;
}
```

- b) Write a test program with a main method that prompts the user to enter two strings and calls the method implemented above.

**Solution:**

```
public static void main(String [] args) {

    Scanner sc = new Scanner(System.in);
    String x = sc.next();
    String y = sc.next();
    System.out.println(checkOccurrences(x,y));
}
```

**Exercise 3**

(12 Marks)

For each of the calls to the following recursive method below, indicate what output is produced:

```
public static void mystery(int x, int y) {
    if (y <= 0) {
        System.out.print("0 ");
    }
    else
        if (x > y) {
            System.out.print(x + " ");
            mystery(x - y, y);
        }
        else {
            mystery(x, y - x);
            System.out.print(y + " ");
        }
}
```

a) `mystery(6, 3);`

**Solution:**

6 0 3

b) `mystery(3, 10);`

**Solution:**

3 2 0 1 4 7 10

**Exercise 4**

(12 Marks)

We would like to count the number of ears in a mutant bunny line assuming that even numbered mutant bunnies have 2 ears and odd numbered mutant bunnies have 3 ears.

Write two methods `mutantBunnyEars` that given the number of mutant bunnies returns the number of ears of the bunnies. Implement the methods once recursively and once sequentially.

For example:

- `mutantBunnyEars(1)` would return 3
- `mutantBunnyEars(2)` would return  $3+2 = 5$
- `mutantBunnyEars(3)` would return  $3+2+3 = 8$

## a) Recursive Method

**Solution:**

```
public static int mutantBunnyEars(int x){
    if(x == 1)
        return 3;
    else
        if(x%2 == 0)
            return 2 + mutantBunnyEars(x-1);
        else
            return 3 + mutantBunnyEars(x-1);
    }
}
```

## b) Sequential Method (without loops nor recursion)

**Solution:**

```
public static int mutantBunnyEarsSeq(int numBunnies){

    int answer = 5 * numBunnies / 2;
    answer += 3 * numBunnies % 2;

    // another solution
    int t = 3*((numBunnies+1)/2) + 2*(numBunnies/2);
    System.out.println(t);

    return answer;
}
```

**Exercise 5**

(15 Marks)

At McDonalds one can buy chicken nuggets in packages containing 6, 9 or 20 pieces. Write a Java method that takes as input an integer, num, and decides about the number of packages of nuggets to buy.

For example,

- nuggets (41) should display:

2 package(s) of 6 pieces, 1 package(s) of 9 pieces and 1 package(s) of 20 pieces

- nuggets (50) should display:

2 package(s) of 6 pieces, 2 package(s) of 9 pieces and 1 package(s) of 20 pieces  
5 package(s) of 6 pieces, 0 package(s) of 9 pieces and 1 package(s) of 20 pieces

- nuggets (14) should display

You cannot buy 14 nuggets

**Solution:**

```
public static void nuggets(int num){
    boolean flag = false;
    for(int a =0 ; a <= num/6+1; a++)

    for(int b=0; b <= num/9+1; b++)

    for(int c = 0; c <= num/20+1; c++)

    if (6*a + 9*b + 20*c == num)
    {
        flag = true;
        System.out.println(a +" package(s) of 6 pieces, "+
        b +" package(s) of 9 pieces and "+
        c +" package(s) of 20 pieces.");
    }

    if(!flag)
        System.out.println("You cannot buy "+num+" nuggets")
    }
}
```

---

**Scratch paper**

---

---

**Scratch paper**

---



---

**Scratch paper**

---