

## Midterm Exam Version 1 (120 points)

**Question 1: (10 pts)**Show the output from the following code segments:

	Code	Output												
A (5)	<pre>int [] numsA = {19, 17, 15, 13, 11}; int [] numsB = {2, 4, 6, 8, 10, 12}; numsA[0] = numsA[4]; numsB[3] = 3 * numsA[0]; numsB[0] = 50; numsA = numsB; if(numsA[5] == 12){     System.out.println("fly"); } else{     System.out.println("flop"); } System.out.println("A : "+Arrays.toString(numsA)); System.out.println("B : "+Arrays.toString(numsB));</pre>													
B (5)	<pre>public static void q1B(char c, int x){     switch(c){         case 'a':         case 'A':             if(x % 2 == 0){                 System.out.print("yummy ");             }             System.out.println("pineapple");             break;         case 'c':             System.out.println("Superman");             break;         case 'b':             System.out.print("Donuts ");         case 'd':             System.out.println("eaten");             while(x&gt;0){                 System.out.println("by alligators");                 x--;             }             break;         default:             System.out.println("parrots");             break;     } }</pre>	<table><tr><th>method call</th><th>output</th></tr><tr><td>q1B('a', 6);</td><td></td></tr><tr><td>q1B('b', 2);</td><td></td></tr><tr><td>q1B('c', 6);</td><td></td></tr><tr><td>q1B('b', 0);</td><td></td></tr><tr><td>q1B('e', 6);</td><td></td></tr></table>	method call	output	q1B('a', 6);		q1B('b', 2);		q1B('c', 6);		q1B('b', 0);		q1B('e', 6);	
method call	output													
q1B('a', 6);														
q1B('b', 2);														
q1B('c', 6);														
q1B('b', 0);														
q1B('e', 6);														

**Midterm Exam Version 1 (120 points)**

**Question 2: (36 pts)** Write a complete class **Person** that has the following attributes, constructors and methods. Use the UML and explanations below.

Person

- numPeople : static int //initialize to 0, increment inside the constructor
- name : String
- id : int
- # generateId() : void //method to increment numPeople and set the id using numPeople
- + Person() //increment numPeople and assign value to id
- + Person(String) //call the default Person constructor
- + getNumPeople() : static int //no setter for numPeople
- + getId() : int //no setter for Id
- + getName() : String
- + setName(String) : void
- + toString() : String
- + equals() : boolean

2a) (3) Write all **variables** as per the UML

2b) (12) Write all the **constructors** as per the UML

Note: Do not call the **generateId** method from within the Person constructor

2c) (3) Write the **generateId** method

Note: Used to generate a replacement id, and is available for inheritance

2d) (8) Write the **getters/accessors** and **setters/mutators** as per the UML

2e) (10) Write the overridden methods from the Object class (**toString** and **equals**)

Note1: The **toString** should be formatted as "Person: Name= %20s | ID= %10d\n".

Note2: The **equals** method should use the **name** and **id** to determine equality.

**Solution Question 2:**

**Midterm Exam Version 1 (120 points)**

Midterm Exam Version 1 (120 points)

## Midterm Exam Version 1 (120 points)

**Question 3: (4 pts)** Write a class named **StudentCapacityException** that inherits from **Exception**. Make sure to define both the **StudentCapacityException()** and **StudentCapacityException(String)** constructors.

**Solution Question 3:**

Use the following interface when answering questions 4 and 5.

```
public interface Academic {  
    void prepareForClass();  
    void prepareForExam();  
}
```

---

**Midterm Exam Version 1 (120 points)**

**Question 4: (15 pts)** Write a complete class **Student** that extends the **Person** class and implements the **Academic** interface provided. Use the UML and explanations below.

Student

- sld : String // is the inherited int id prefixed by the String "S" as shown "**S**" + id
- numCourses : int
- courses [ ] : String
- + toString() : String

**NOTE:** Assume all other necessary constructors, getters, setters, and other methods have been written already even though they are not shown here (**do not rewrite them**)

**4a) (3)** Write a complete class **Student** that extends the **Person** class and implements the **Academic** interface provided.

**4b) (3)** Write all **variables** as per the UML

**4c) (3)** Write one of the overridden method from the **Person** class (**toString**)

Note: Remember to include the functionality of the parent class's toString() method. In addition to parent class's functionality, the output should have:

"Student: sID= %10s | numCourses= %10d\n".

**4d) (3)** Implement the **prepareForClass()** method from the **Academic** interface so that it prints out "Doing homework and battling monsters for many hours"

**4e) (3)** Define the **void prepareForExam();** method from the **Academic** interface so that it prints out "Studying diligently but freaking out!"

**Solution Question 4:**

**Midterm Exam Version 1 (120 points)**

**Midterm Exam Version 1 (120 points)**

**Question 5: (44 pts)** Write a complete class **Faculty** that extends the **Person** class and implements the **Academic** interface provided. Use the UML and explanations below.

Faculty

- fld : String // is the inherited int id prefixed by the String "F" as shown "F" + id
- numStudents : int // the current number of students in Students[ ] array
- courses [ ] : String //default length of 4
- students [ ] : Student //default and max length of 15
- # generateId() : void //call parent's getId() and prefix it with "F" to assign to fld "F" + id
- + Faculty(String, String [ ])
- + Faculty(String, String [ ], Student [ ])
- + addStudent(Student) : void throws StudentCapacityException
- + getNumStudents() : int
- + getStudentsAsString() : String // single String, student names space separated
- + getFld() : String

**5a) (3)** Write all **variables** as per the UML

**5b) (18)** Write all the **constructors** and the **generateId** method as per the UML

Note1: Remember to use the **generateId** method from within the Faculty constructor

Note2: If the passed in Student array is longer than 15, copy only the first 15 students

**5c) (7)** Write the **getters/accessors** and **setters/mutators** as per the UML

**5d) (10)** Write the **addStudent** method so that it adds the Student to the array and increments **numStudents**. If the capacity of 15 is exceeded throw the **StudentCapacityException** instead.

**5e) (3)** Define the **prepareForClass()** method from the **Academic** interface so that it prints out "Prepare lecture material, exercises and good examples for class"

**5f) (3)** Define the **void prepareForExam();** method from the **Academic** interface so that it prints out "Make a fair exam and prepare solutions"

---

**Solution Question 5:**



**Midterm Exam Version 1 (120 points)**

**Midterm Exam Version 1 (120 points)**

## Midterm Exam Version 1 (120 points)

**Question 6: (11 pts)** Write the code to create instances of Person, Faculty, Student and invoke their methods as specified below.

a	Person <b>p1</b> without specifying the name
b	Person <b>p2</b> whose name is "Frank"
c	Faculty <b>f1</b> whose name is "Sherman" and teaches "CHE222" and "BIO431"
d	Faculty <b>f2</b> whose name is "Ellen" and teaches "DAN123" and "CMP326"
e	Student <b>s1</b> whose name is "Giannis" and takes "BUCKS" and "DNK123"
f	Show the invocation of <b>toString()</b> on <b>p1</b>
g	Show the invocation that would check the if <b>p1</b> and <b>p2</b> are <b>equal</b>
h	Invoke the <b>getStudentsAsString()</b> method on <b>f2</b>
i	Invoke the <b>prepareForExam()</b> method on <b>f1</b>
j	Invoke the <b>prepareForClass()</b> method on <b>s1</b>
k	Invoke the <b>addStudent(Student)</b> method on <b>f1</b> passing in <b>s1</b> and using exception handling

**Solution Question 6:**

**Midterm Exam Version 1 (120 points)**