

Name \_\_\_\_\_ Period \_\_\_\_\_

**Skill 18.01 Exercise 1**

Refer to the class below.

```
public class Student{  
    private String name;  
    private int gradeLevel;  
    private double GPA;  
    private boolean hasScholarship;  
  
    public Student(){  
  
    }  
    public void setName(String n){  
        name = n;  
    }  
    public String getName(){  
        return name;  
    }  
}
```

Complete the stack and heap diagram for the following calls, then indicate the output that would be printed.

```
Student student1 = new Student();  
String someName = "Bob";  
Student1.setName(someName);  
Student student2 = new Student();  
Student2.setName("Marvin");  
Student1 = student2;  
System.out.println(student1.getName());  
System.out.println(student2.getName());
```

Stack	Heap	Output

Name \_\_\_\_\_ Period \_\_\_\_\_

**Skill 18.02 Exercise 1**

The following method was added to the Student class above,

```
public int getGradYear(){  
  
    int gradYear = 0;  
    int year = YearMonth.now().getYear();  
    int month = YearMonth.now().getMonthValue();  
    if(month>=6){  
        gradYear = 12 - gradeLevel + year + 1;  
    }else{  
        gradYear = 12 - gradeLevel + year;  
    }  
    return gradYear;  
}
```

Student usernames are based on the students first initial followed by their graduation year. For example, a student name “Bart” is in grade 10. So, his username is b2023. Write the getUsername method below which returns the username of a student.

**Skill 18.03 Exercise 1**

Refer to the Student class above. Write the method getHasScholarship, which returns true if a student has a GPA over 3.5 and false otherwise.

Name \_\_\_\_\_ Period \_\_\_\_\_

**Skill 18.04 Exercise 1**

Consider the following class declarations

```
public class SumNums{
    private int num1;
    private int num2;

    public SumNums(int a, int b){
        int sum = a + b;
    }
    public int getSum(){
        return sum;
    }
    public int reverseNum(int num){
        int reversed = 0;

        while(num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
    }
    public int anotherMethod(int num){
        return reversed*Math.pow(reversed, num);
    }
}
```

The code above has errors. Fix the code so it works as intended. .

Name \_\_\_\_\_ Period \_\_\_\_\_

**Skill 18.05 Exercise 1**

Consider the following partial class declaration

```
public class Student{
    public String name;
    private int gradeLevel;
    private double GPA;
    private boolean hasScholarship;

    public Student(){

    }
    public void setName(String n){
        name = n;
    }
    public String getName(){
        return name;
    }
}
```

The following declaration appears in another class. For each line of code, indicate whether or not it will compile without error. If it does not compile indicate why.

```
Student student1 = new Student();
Student student2 = new Student();
```

```
student1.name = "Bart";
```

```
System.out.println(student1.getName());
```

```
System.out.println(student1.setName("Kyle"));
```

```
System.out.println(student1.GPA);
```

```
student1 = student2;
```

```
System.out.println(student1.getName());
```

```
String result = student2.setName("Homer");
```

```
System.out.println(result);
```