CSCE 145 Exam 1 Review Answers

This exam totals to 100 points. Follow the instructions. Good luck!

Chapter 1
This chapter was mostly terms so expect a fill in the blank style questions on definition. Remember look for those red, italics, bold, underlined words from the first slides. Here are some definitions that may be on the exam.

1. A **bit** is a digit with a value of either 0 or 1.
2. A **byte** consists of 8 bits.
3. Each byte in main memory resides at a numbered location called its **address**.
4. A **program** is a set of instructions for a computer to follow.
5. A **compiler** translates a program from a high-level language to a low-level language the computer can run.
6. A **package** is a library of classes that have been defined already. Such as

   ```java
   import java.util.Scanner;
   ```

7. A **variable** is used to store data such as number and characters, and has a type and a name associated with it.
8. Java translates a program into **byte-code** before compiling it to machine code.
9. A Java program typically consists of several pieces of code called **classes**.
10. An **algorithm** is a well defined set of instructions for solving a problem.
11. Algorithms usually are expressed in English or in **pseudocode**.
12. The grammar rules for a programming language are called the **syntax** of the language.
13. The three types of errors are **syntax**, **run-time**, and **logic**.
14. The place where the computer starts a running instructions is called the **entry point**. In java it is

   ```java
   public static void main(String[] args)
   {
       //Starts here!
   }
   ```

15. An instruction to the computer is called a **statement**, and it ends with a semicolon (;

16. Java, C#, C++, Visual Basic, Python, Ruby are all examples of **high-level** languages, which is easier to write and understand than machine code or assembly.

Chapter 2
This chapter was solely on variables. You may expect terminology, short answer, and programming, but probably not all 3.

Terminology Questions
1. When you **declare** a variable, you provide its name and type.
2. A **primitive type** is used for simple, non-decomposable values such as an individual number or individual character. Like (int, double, char)
3. A **class type** is used for a class of objects and has both data and methods. Like (String, Scanner, Random)

4. **Initializing** a variable is where you assign the variable a value before it is ever used. This is good programming practice as it doesn’t rely on a default value.

5. An **identifier** is a name, such as the name of a variable.

6. The equals sign (=) is called the **assignment operator**.

7. Words such as **if** are called **keywords or reserved words** and have special, predefined.

8. Uninitialized primitive variables may have a **default value**.

Short Answer Questions – These questions are more conceptual and could be true-or-false questions with a short explanation, or write what the program snippet will print out. Also you can expect a question on the **mod (%)** operator.

1. Will this programming snippet give an error? True or False? If this does cause an error then why?

   ```java
   String value1 = "34.0";
   int value2 = value1;
   ```

   Answer: True, it will cause an error because you cannot assign a String class type to an integer primitive type.

2. What will this program print out?

   ```java
   int number = 20%7
   System.out.println(number);
   ```

   Answer: 6, 20/7 is 2 with a remainder of 6. Remember mod gives the remainder.

Programming Questions – These questions will be very similar to the lab assignments or the homework.

1. Write a program that prompts the user for a number which corresponds to degrees and change that number into radians. Then prints both the input and result. This conversion is rad = deg x 3.14159/180.0. Also assume that the input and the result are both double.

   ```java
   import java.util.Scanner;
   public static void main(String[] args)
   {
       Scanner keyboard = new Scanner(System.in);
       //Put your code here

       System.out.println("Enter some degrees");
   }```
double input = keyboard.nextDouble(); // get the input
double result = input*3.14159/180.0; // calculate the result

// print the result
System.out.println(input+" degrees is "+result+" radians");

Chapter 3
This chapter dealt with branch statements: if, else if, else, and switch. Not much
terminology is here so there will be questions mainly on short answer and possibly
programming. Expect a conversion from either switch to if, else if, else or the other way
around.

Short Answer Questions
1. When this program is executed what will it print out

```java
int number = 10;
if(number < 5)
{
    System.out.println("A");
}
else if(number < 100 && number > 5)
{
    System.out.println("B");
}
else if (number < 20)
{
    System.out.println("C");
}
else
{
    System.out.println("D");
}
if(number > 1 || number <=5)
{
    System.out.println("E");
}
else if(number > 2 && number <=40)
{
    System.out.println("F");
}
else
{
    System.out.println("G");
}
```
Programming Questions
If I do put one of these expect something along the same lines as labs, homeworks, or examples

1. Write a program where the user enters an integer and the program checks whether or not it is between 0 and 255 (both inclusive so it can be 0 and 255). If it is then the number is valid and the program will say that the number entered is valid. Otherwise it will prompt the user that they have entered an invalid number.

```java
import java.util.Scanner;
public static void main(String[] args)
```
```java
Scanner keyboard = new Scanner(System.in);
//Put your code here

System.out.println("Enter a number between 0 and 255 inclusively");
int number = keyboard.nextInt(); //get the input
if(number>=0 && number<=255) //this is valid
{
    System.out.println("This is a valid number!");
}
else //this is not valid
{
    System.out.println("The input is invalid");
}
```

Chapter 4
This chapter dealt with loops: while, for, and do while. This will mostly be short answer and programming.

Short Answer
Be able to explain the difference between “do-while” and “while”. Also be able to convert a “for” into a “while” and vice versa. Finally, be able to figure out what a loop prints out.

1. Explain the difference between a while and a do-while loop?
   The body of a do while will run at least once, and a while’s body may never run at all.
2. Convert this for statement into a while statement

```java
for(int i=0;i<20;i++)
{
    System.out.print(i+" ");
}
```

Answer:
```java
int i=0;
while(i<20)
{
    System.out.println(i+" ");
    i++;
}
3. What does the following program display?

```java
for(int i=0;i<10;i++)
{
   System.out.print(2*i+" ");
}
```

Answer:
0 2 4 6 8 10 12 14 16 18

Programming Questions
Similar to the labs and examples. Also expect a question either on this exam or the final that has you print out something like the triangle from the lab assignment.

1. Write a program where the user enters a number, and the program prints out right triangle where the number is the width of the base. The triangle’s base must be a the top and then go to the points.
   Example: If the user enters 5 the program will print out
   
   * * * * *
   * * * *
   * * *
   * *

   ```java
   import java.util.Scanner;
   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      //Put your code here

      System.out.println("Enter the size of the triangle");
      int number = keyboard.nextInt(); //get the base size
      for(int i=number;i>0;i--) //this one counts backwards
      {
         //each i corresponds to the number of * to be printed
         for(int j=i;j>0;j--)
         {
            System.out.print("*");
         }
         System.out.println(); //next line
      }
   }
   ```
2. Write a program where the user enters a number, and then the program adds 10 from that number until the number is greater than 50. It must print out the new value at each step.
Example if the user enters 14 the program will print out

```
14
24
34
44
```

```
import java.util.Scanner;
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    //Put your code here
    System.out.println("Enter a number");
    int number = keyboard.nextInt(); //get the initial number

    while(number <= 50) //until that number is greater than 50
    {
        System.out.println(number);
        number+= 10; //keep adding 10
    }
}
```


```java
{ 
    min = a[i]; //reset the min to the current index
}
if(a[i]>max) //if the current index is greater than the max
{
    max = a[i]; //reset the max to the current index
}
}
int result = max-min; //find the result
System.out.println(max+-+min+=?=result); //print it out
}

2. Write a program that multiplies all the number in an array and then prints out the result.
Example if the array given is {2,4,6,8} the program would print 384
public static void main(String[] args)
{
    Scanner keyboard = new Scanner(System.in);
    int[] a = {2,4,6,8};
    //Put your code here
    int result = a[0]; //keeps track of the results
    for(int i=1;i<a.length;i++)
    {
        result *= a[i]; //adds the current index to the result
    }
    System.out.println(result); //prints the result
}

3. Write a program that goes through an array and then changes every instance of an even number into a 0, and then prints out the resulting array.
Example if the array given is {1,2,3,4,5,6,7,8} the program will print out 1 0 3 0 2 0 7 0
public static void main(String[] args)
{
    Scanner keyboard = new Scanner(System.in);
    int[] a = {1,2,3,4,5,6,7,8};
    //Put your code here
    //Looks for the even numbers in a loop
    for(int i=0;i<a.length;i++)
    {

//it is even if it is divisible by 2 and the remainder is 0
if(a[i]%2 == 0)
{
    a[i] = 0;
}

//Prints the results
for(int i=0;i<a.length;i++)
{
    System.out.print(a[i]+" ");
}

4. Will this program cause an error? If so why does it cause an error? If not show what the program prints out?

    int[] a = {1,2,3,4,5};
    for(int i=0;i<=5;i++)
    {
        System.out.println(a[i]);
    }

Answer: This program will cause an error. When it reaches index 5 or a[5] this will go out of the boundary of the array, and cause an out of bound exception.