Students may use their text book, any hand written notes, any digital notes or programs written by the student (You may use your flash drive), and the internet.

Thursday, March 17th – 2016

8:00am – 11:00am – Be sure to attempt questions or programming tasks. Partial Credit will be awarded.

**SHORT ANSWER SECTION**

1. **(5 Points) Name 5 different code blocks we covered in class to this point.**

Answer:- The code blocks always have two curly braces, one starting curly braces and another ending. Five different code blocks are,

1. main code block
2. while loop code block
3. if/ else statement code block
4. for loop code block
5. switch statement code block
6. **(10 points) a) Describe the purpose of a loops in programming.**

Answer:- Looping is the process of running same block of code until the condition met that we have specified in the condition statement. It helps to run the same block of code as many times as we desire. This way we don’t have to write same block of code many times, which saves time, space and easy for debugging and understanding the program.

b) **Compare and Contrast indefinite and definite loops. Be as detailed as possible**.

Indefinite Loop- Indefinite looping keeps looping infinitely but it has stops at some point.

Definite Loop- Unlike indefinite looping, definite loops stops executing the code block after certain time period as we have stated in the condition. For the definite loop, condition statement is needed as we want stop executing the program after certain condition as we desire.

1. **(5 Points) How does *char* relate to Strings in Java.**

Char- character is a primitive data type, and a single character.

String in Java- are objects, and can have zero to multiple characters. It occupy more memory blocks.

- character can have only one character while string can have character from zero to multiple characters. Char uses single quotation while string use double quotations.

1. **(10 Points) – a) Using || to connect multiple Boolean expressions, does it lessen or widen the scope of the Boolean Expression returning a true value? Explain in as much detail as possible.**

Answer:- Yes, it widen the scope of the Boolean Expression returning a true value. In || (or), to be true, any at least one of the condition stated in the statement must be true. This way it widen the scope of the Boolean Expression.

e.g. if (true || false || false || false);

it will return true value.

It is different than && ( and), while in && all conditional statement must be true to return the true value.

b) **Using && to connect multiple Boolean expressions, does it lessen or widen the scope of the Boolean Expression returning a true value? Explain in as much detail as possible.**

Answer: - In contrast to ||, && lessen the scope of the Boolean Expression returning a true value. To be true for this, all conditional statement must be true, otherwise it will return the false value.

e.g if (true || false || false || false);

it will return false value.

To be true all conditional statement must be true like

if (true || true || true || true);

it will return true value

1. **(5 Points) Give a couple examples of Java programming conventions. Give 3 examples of Java Syntax rules.**

Answer: - Convention is a set of guidelines for a specific programming language that recommend programming style, practice and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, declarations, statements, white space, naming conventions etc.

3 Java Syntax rules.

1. We should declare the type while declaring variable name
2. Variable name should not have space/ staring with numbers
3. Each line should be ended with semicolon.
4. We declare the variable before we use it.

6. (5 Points)

**Write an English sentence that has correct syntax but incorrect *semantics***.

Syntax refers to a grammatically correction in English language, while semantics means its meaning.

e.g. The green apple ate a juicy bug.

This is syntactically correct but doesn’t have meaning.

**Write an English sentence that has correct semantics but incorrect *syntax*.**

e.g. I ate rice

here no period at the end of sentence, which is correct semantics and incorrect syntax.

**SHORT ANSWER SECTION Cont…**

5. a) (10 points)

**Complete Logic Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **P** | **S** | **P &S** | **P | S** | **P ^ S** | (P&S) && (P|S) | !S || False || !P |
| **FALSE** | **TRUE** | **False** | **True** | **True** | **False** | **True** |
| **FALSE** | **FALSE** | **False** | **False** | **False** | **False** | **True** |
| **TRUE** | **FALSE** | **False** | **True** | **True** | **False** | **True** |
| **TRUE** | **TRUE** | **True** | **True** | **False** | **True** | **False** |

**BASIC SYNTAX ERRORS:** (10 Points – 2 Points Each) Spot the basic Syntax errors

1. **System.out.println(“Print your name);**

Answer:- Missing double quotation (“) at the end of the string. So the correct for will be,

System.out.println(“Print your name”);

2) **public class Constructor {**

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

**System.outprintln();**

**}**

Answer: - There is 2 thing missing, one is curly braces at the end of the program, other is dot(.) after out. The correct form is,

public class Constructor {

public static void main(String[] args) {

System.out.println();

}

}

3) **int a;**

**int = 0**

Answer: - in the second one, missing variable name and semicolon(;) at the end. The correct form will be,

Int a = 0;

4) **Public static void main(String[] args)**

Answer:- two things are wrong, ‘P’ is capitalized and missing curly braces.

The correct form will be

public static void main(String[] args) {

}

5**) int b;**

**b = 1.5;**

Answer: - data type (int) is wrong. The data type should be float. The correct form will be

float b;

float b = 1.5;

**SHORT PROGRAMMING TASKS:**

(15 points) TASK 1**: (Using &&, ||, ^ operators) Write a program that prompts the user to enter an integer and determines whether it is divisible by 5 and 6, whether it is divisible by 5 or 6, and whether it is divisible by 5 or 6, but not both.**

**Sample Output**

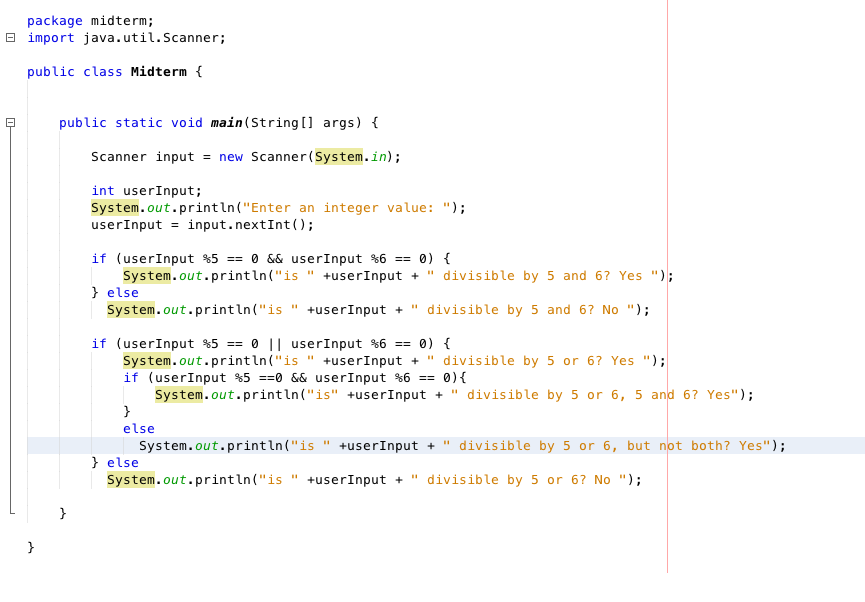
**Enter an integer:**

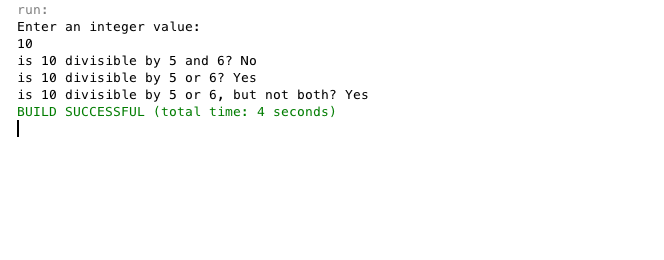
**10 // Users enters 10**

**Is 10 divisible by 5 and 6? No**

**Is 10 divisible by 5 or 6? Yes**

**Is 10 divisible by 5 or 6, but not both? Yes**





(15 points) TASK 2: **Update Task 1:**

**Place the above code into a do-while / switch case menu.**

**Allow the user to choose which operation she will perform.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Main Menu: \***

**\* Enter # to run program or Quit \***

**\* 1)** Is Input divisible by 5 and 6 **\***

**\* 2)** Is Input divisible by 5 or 6 **\***

**\* 3)** In input divisible by 5 or 6, but not both  **\***

**\* 4) Quit \***

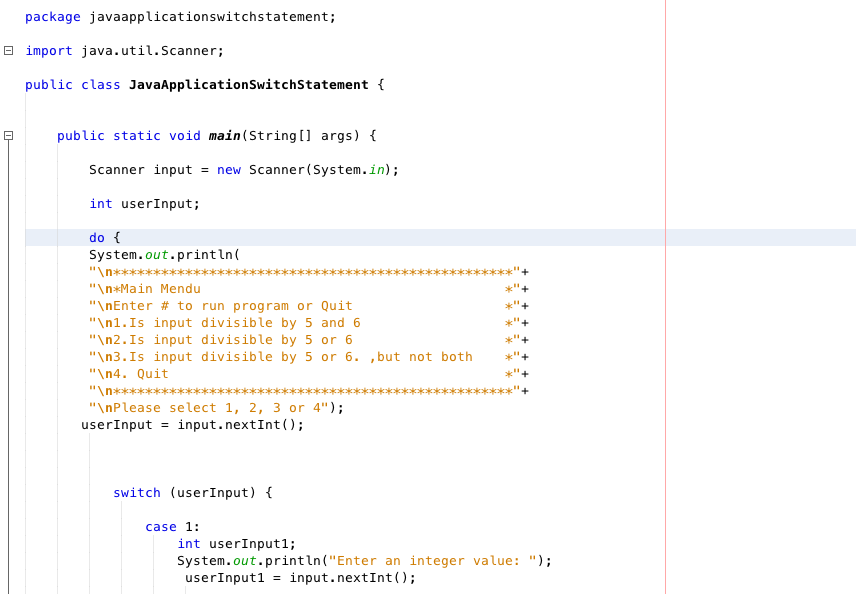
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1)

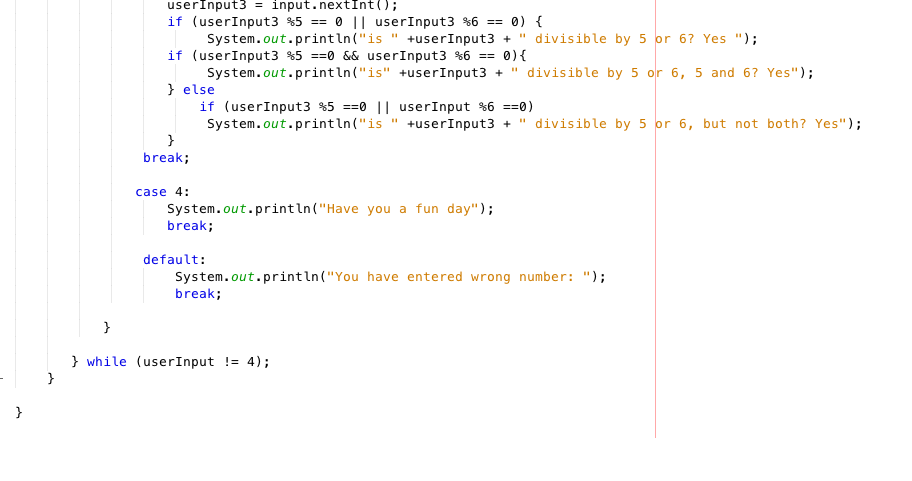
Enter an integer:

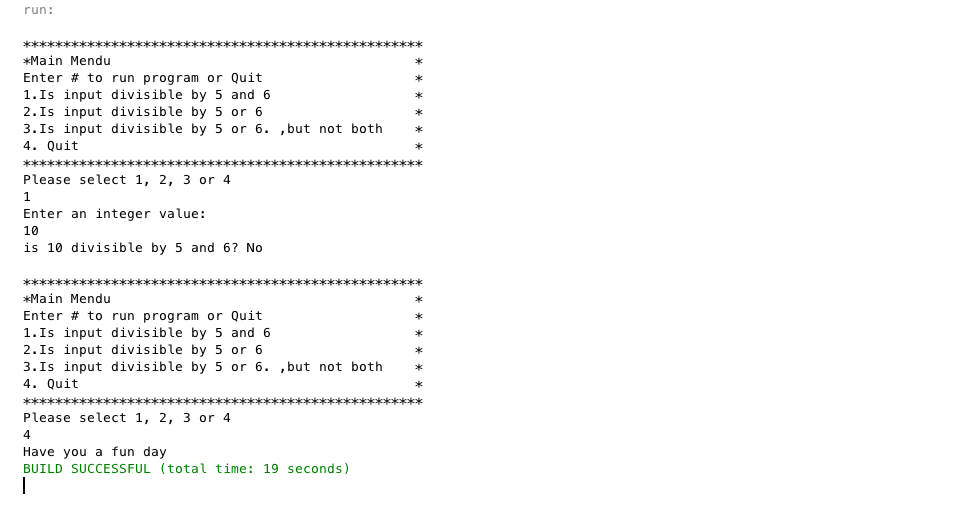
10 // Users enters 10

Is 10 divisible by 5 and 6? No



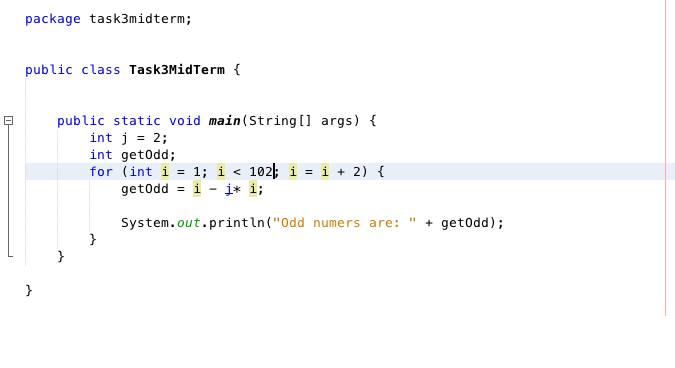




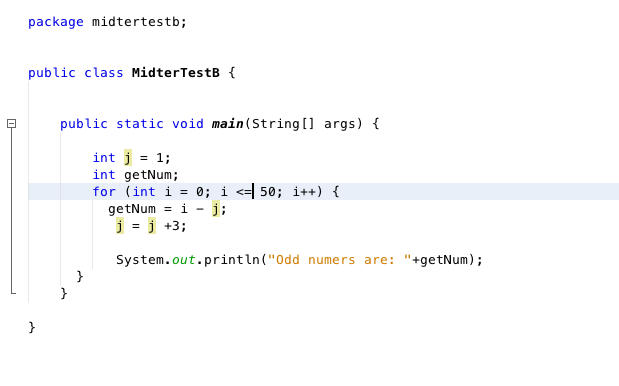


(30 points) TASK 3: Use differentloops to **print the odd / negative numbers 1 to 101.** All programs will print the same output in the same order.

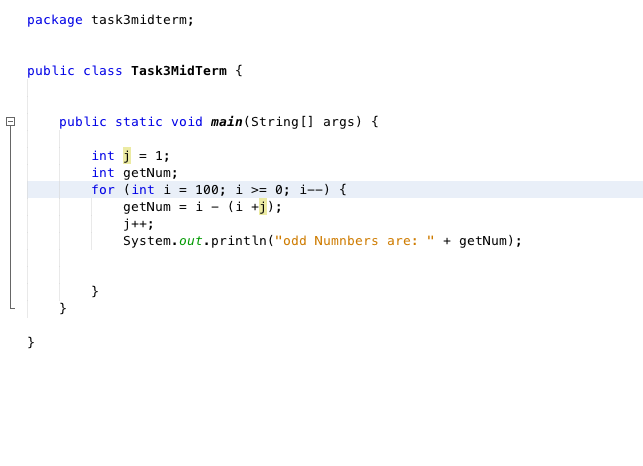
1. Using a **for** loop that increments the loop control variable by 2 each iteration



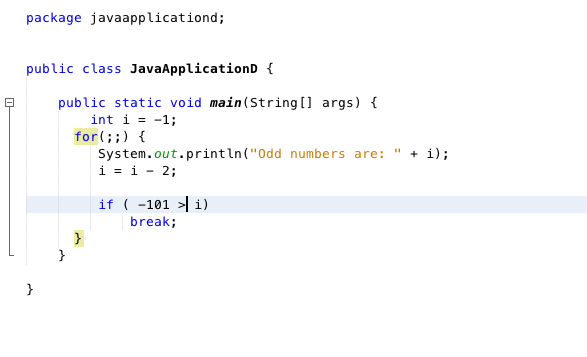
1. Using a **for** loop whose loop control variable goes from 0 to 50.



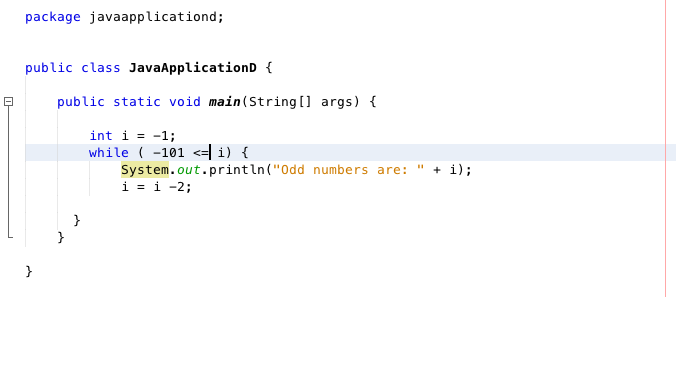
1. Using a **for** loop whose loop control variable goes from 100 down to 0.



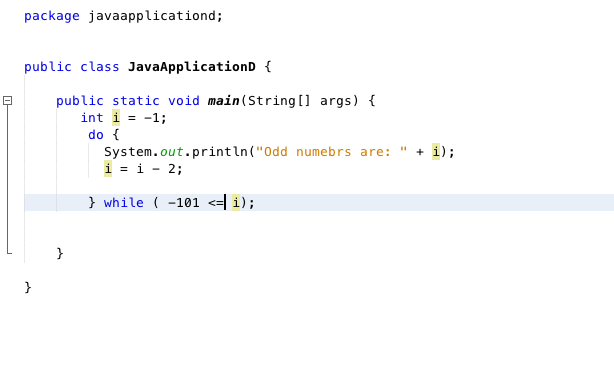
1. Using an infinite **for** loop with no conditional expression and exiting the loop with a **break** statement.



1. Using a **while** loop.



1. Using a **do-while** loop.



There should be 6 different Snipping photos. One photo for each program A – F.

-1

-3

-5

-7

…

-101

output for all program:

