

# Programs written in class in the course CSC2302 (Computer Programming II) during the second semester 2022/2023 academic session

## Program 1

```
/* This program demonstrate how to write simple Java program with some
variables
* having fixed values. Here, value for the
* radius of the circle is fixed to 2.0
* File name: ComputeCircleArea.java
* */
// Beginning of the class
public class ComputeCircleArea {

    public static void main(String[] args) {
        //variable declaration
        double radius = 2.0;
        double area = Math.PI * radius * radius;
        System.out.println("The area of the circle is: " + area);
    }
} // End of the class
```

## Program 2

```
/* This program lets the user input the
* radius of the circle instead of fixing it
* to a given value. The program uses a
* Scanner class from the java.util package
* to read input from the console
* */

import java.util.Scanner;

// beginning of the class. Filename: ComputeCircleAreaWithInput.java
public class ComputeCircleAreaWithInput {

    // beginning of main method
    public static void main(String[] args) {
        // create Scanner object for reading input
        Scanner input = new Scanner(System.in);

        // ask the user for the radius
        System.out.println("Pleas, enter the radius");
        double radius = input.nextDouble();

        //now, compute the area
        double circleArea = Math.PI * radius * radius;
```

```

        // display the result
        System.out.println("The area of the circle is: " + circleArea);
    } // end of main method
} // end of the class

```

### Program 3

```

/* This program demonstrates how to use conditional statements in a
program.
* The program asks a user to enter a student's mark and then prints an
* appropriate message for the student.
* */

```

```

import java.util.Scanner;

```

```

// Filename: UseOfConditional.java
// Beginning of the class.

```

```

public class UseOfConditional {

    public static void main(String[] args) {
        // create a Scanner object
        Scanner scan = new Scanner(System.in);

        // get the score from the user
        System.out.println("Enter the student's score");
        double examScore = scan.nextDouble();

        // declare a variable to hold the grade
        char grade = ' ';

        // ensure that the score is not negative or greater than 100
        if(examScore < 0.0 || examScore > 100) {
            System.out.println("You have entered an invalid score");
            return;
        }
        else {
            if(examScore >= 70)
                grade = 'A';
            else if(examScore >= 60)
                grade = 'B';
            else if(examScore >= 50)
                grade = 'C';
            else if(examScore >= 45)
                grade = 'D';
            else if(examScore >= 40)
                grade = 'E';
            else
                grade = 'F';
        }
    }
}

```

```

        // print the grade
        System.out.println("The grade is " + grade);
    }
}

```

#### Program 4

```

/* This program demonstrates how to use switch statements in a program.
 * The program asks a user to enter a student's mark and then prints an
 * appropriate message for the student.
 * */

```

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

```
// Filename: UseOfSwitchStatement.java
```

```
// Beginning of the class.
```

```
public class UseOfSwitchStatement {
```

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

```
        // create a Scanner object
```

```
        Scanner scan = new Scanner(System.in);
```

```
        // get the score from the user
```

```
        System.out.println("Enter the student's score");
```

```
        int examScore = scan.nextInt();
```

```
        // declare a variable to hold the grade
```

```
        char grade = ' ';
```

```
        // ensure that the score is not negative or greater than 100
```

```
        if(examScore < 0 || examScore > 100) {
```

```
            System.out.println("You have entered an invalid score");
```

```
            return;
```

```
        }
```

```
        else {
```

```
            switch(examScore / 10) {
```

```
                case 10:
```

```
                case 9:
```

```
                case 8:
```

```
                case 7: grade = 'A'; break;
```

```
                case 6: grade = 'B'; break;
```

```
                case 5: grade = 'C'; break;
```

```
                case 4:
```

```
                    if(examScore >= 45) grade = 'D';
```

```
                    else grade = 'E'; break;
```

```
                default: grade = 'F';
```

```
            }
```

```
        }
```

```
        // print the grade
```

```
        System.out.println("The grade is " + grade);
```

```
    }  
} // end of the class
```

### Program 5

```
/* This program demonstrates how to use a conditional expression to find  
 * the biggest value between 2 numbers  
 * */
```

```
import java.util.Scanner;  
// Filename: UseOfConditionalExpression.java  
// Beginning of the class  
public class UseOfConditionalExpression {  
  
    public static void main(String[] args) {  
        // create a Scanner object  
        Scanner scan = new Scanner(System.in);  
  
        // get the first number from the user  
        System.out.println("Enter the first number");  
        double num1 = scan.nextDouble();  
  
        // get the second number from the user  
        System.out.println("Enter the second number");  
        double num2 = scan.nextDouble();  
  
        // determine the largest using a conditional expression  
        double maxValue = num1 > num2 ? num1 : num2;  
  
        // display the largest value  
        System.out.println("The largest value is " + maxValue);  
    }  
} // end of the class
```

### Program 6

```
/* This program solves a quadratic equation using the formula method.  
 * the coefficients are supplied by the user at runtime using Scanner  
object  
 * */
```

```
import java.util.Scanner;  
  
//Filename: QuadraticSolver.java  
//Beginning of the class.  
public class QuadraticSolver {
```

```

public static void main(String[] args) {
    // declare the coefficients a, b, and c
    float a, b, c;

    // declare a variable called d to hold b*b - 4*a*c
    float d;

    //declare the roots
    double x1, x2;

    //get inputs
    Scanner input = new Scanner(System.in);
    System.out.println("Enter value of a");
    a = input.nextFloat();
    System.out.println("Enter value of b");
    b = input.nextFloat();
    System.out.println("Enter value of c");
    c = input.nextFloat();
    d = b * b - 4 * a * c;
    if(d > 0) {
        x1 = (-b + Math.pow(d, 0.5))/(2 * a);
        x2 = (-b - Math.pow(d, 0.5))/(2 * a);
        System.out.println("The distincts roots are: " + x1 + ", "
+ x2);
    }
    else if(d == 0) {
        x1 = x2 = -b/(2 * a);
        System.out.println("The equal roots " + x1 + ", " + x2);
    }
    else
        System.out.println("The equation has imaginary roots");
}

} // end of class

```

### Program 7

```

/* This program will print Welcome to CSC2302 200 times
*/

// Filename: UseOfWhileLoop.java
// beginning of the class
public class UseOfWhileLoop {

    public static void main(String[] args) {

        int count = 0; // a counter variable
        while(count++ < 200)
            System.out.println(count + " Welcome to CSC2302");
    }
}

```

```
    }  
} // end of the class
```

### Program 7

```
/* This program sums up the sequence 1, 2, 3,...,100  
 * using a while statement. You can also use for or do...while to achieve  
 * the same result.  
 * */
```

```
// Filename; Sum100Integers.java  
public class Sum100Integers {  
  
    public static void main(String[] args) {  
  
        int i = 1, sum = 0;  
        while(i <= 200)  
            sum += i++;  
  
        //display the final sum  
        System.out.println("The sum is: " + sum);  
    }  
}
```

### Program 8

```
/* This program adds up a series of values to be entered interactively  
 * by the user via the console. The input stops when a user enters 0.0.  
 * This is an example of the use of sentinel to terminate an infinite  
 loop.  
 * */
```

```
//Filename: RepeatedAddition.java  
  
import java.util.Scanner;  
  
public class RepeatedAddition {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        double val, sum = 0.0;  
  
        //use of do...while loop  
        do {  
            System.out.println("Enter a value, 0 to stop");  
            val = scanner.nextDouble();  
            sum += val;  
  
        }while(val != 0.0);  
    }  
}
```

```

        // display the final sum
        System.out.println("The sum of the numbers entered is: " +
sum);
    }
}

```

### Program 9

```

/*This program demonstrates the use of for loop construct
 * to add up the numbers 1, 2, 3,...,10
 * */

```

```

//Filename; UseOfForLoop.java
public class UseOfForLoop {

    public static void main(String[] args) {

        int sum = 0;
        for(int i = 1; i <= 10; ++i)
            sum += i;

        //display the final sum
        System.out.println("The final sum is: " + sum);
    }
}

```

### Program 10

```

/*This program prints all odd integers between 1 and 200.
 * It uses a for loop
 * */

```

```

//Filename: PrintOddIntegers.java
public class PrintOddIntegers {

    public static void main(String[] args) {
        for(int i = 1; i <= 200; i++) {
            if(i % 2 == 0) continue; // skip even integers
            System.out.print(i + " ");
        }
    }
}

```

### Program 11

### Program 12

```

/*This program prints the multiplication table
 * */

```

```

//Filename: MultiplicationTable.java

```

```

public class MultiplicationTable {

    public static void main(String[] args) {
        for(int i = 1; i <= 12; ++i) {
            for(int j = 1; j <=12; ++j) {
                System.out.print(i * j +"\t");
            }
            System.out.println();
        }
    }
}

```

//End of the class

### Program 13

```

/* This program declares an array of 10 exam scores of students.
 * The program computes and prints the average of the scores.
 * The scores are entered interactively via the console.
 * */

```

```

import java.util.Scanner;

```

```

//Filename: ArrayOfStudentsScores.java

```

```

public class ArrayOfStudentsScores {

    public static void main(String[] args) {
        // create a Scanner object.
        Scanner scanner = new Scanner(System.in);

        //declare the array
        double[] scores = new double[10];

        double sum = 0.0;
        //get the scores from the user.
        for(int i = 0; i < scores.length; ++i) {
            System.out.print("Enter a score");
            scores[i] = scanner.nextDouble();

            //add the score to the sum
            sum += scores[i];
        }

        //print the average score
        System.out.println("The average score is: " +
sum/scores.length);
    }
}

```

### Program 14



```

/* This program creates an array of 100 elements.
 * Each of the elements is initialised to the square of
 * its index, i.e., element at index i is set to i * i
 * */

//Filename: ArraysInitialisedToIndexSquared.java
public class ArraysInitialisedToIndexSquared {

    public static void main(String[] args) {

        //declare and initialise the elements of the array.
        int[] array = new int[100];
        for(int i = 0; i < array.length; i++)
            array[i] = i * i;

        //display the elements of the array.
        for(int ele: array)
            System.out.print(ele + " ");

    }
}

```

### Program 15

```

/*This program collects set of values in an array of size 100,
 * prints the elements in reverse order.
 * */
import java.util.Scanner;

//Filename: ArrayElementsReversedPrinted.java
public class ArrayElementsReversedPrinted {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        double[] array = new double[10];

        //get values in the array
        for(int i = 0; i < array.length; i++) {
            System.out.println("Enter a numerical value");
            array[i] = input.nextDouble();
        }

        //print the elements in reverse order
        for(int i = array.length-1; i >= 0; --i)
            System.out.print(array[i] + " ");

    }
}
//End of the class

```

### Program 16

```

/*This program demonstrates the use of static method to
 * perform a given task. In this case, the method will
 * sum up numbers between first and last elements inclusive.
 * */

public class UseOfStaticMethod {

    public static void main(String[] args) {
        //call the sumNumbers method to add up numbers
        //between 1 and 10
        System.out.println("The sum is " + sumNumbers(1, 10));
    }

    //create a static method
    public static int sumNumbers(int firstNumber, int lastNumber) {

        int sum = 0;
        for(int i = firstNumber; i <= lastNumber; ++i)
            sum += i;

        //return the result to the caller
        return sum;
    } // end of sumNumbers method.
} //end of class

```

## Program 17

```

/*This program reads a sentence and prints the characters
 * in the given sentence in reverse order. Remember, the end
 * of the sentence is reached when a user presses an enter key.
 * This program demonstrates some string processing operations.
 * */

```

```

import java.util.Scanner;

//Filename: TextCharactersReversedPrinted.java
public class TextCharactersReversedPrinted {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.println("Enter a sentence");
        String sentence = scan.nextLine();

        //convert the sentence into an array of characters.
        char[] sentenceArray = sentence.toCharArray();
    }
}

```

```

//print the characters in the sentence in reverse order
for(int i = sentenceArray.length - 1; i >= 0; i--)
    System.out.print(sentenceArray[i]);

System.out.println(); //insert one line space
//print characters of the sentence at even positions
for(int i = 0; i < sentenceArray.length; i++)
    if(i % 2 == 0)
        System.out.print(sentenceArray[i]);

System.out.println(); //insert one line space

// capitalise and print all characters in the sentence
for(int i = 0; i < sentenceArray.length; i++)
    System.out.print(Character.toUpperCase(sentenceArray[i]));

System.out.println();//insert one line space

// capitalise and print all characters at even position
// in the sentence
for(int i = 0; i < sentenceArray.length; i++)
    if(i % 2 == 0)

System.out.print(Character.toUpperCase(sentenceArray[i]));
}
}

```

## Program 18

```

/*This program will print all the prime numbers between 1 and 1000.
 * a prime number is a positive integer greater than 1 that has two
 * factors: 1 and itself.
 * The program uses the method isPrime to determine if a given integer
 * n is prime or not.
 * */

//Filename: PrimeNumbers.java
public class PrimeNumbers {

    public static void main(String[] args) {

        //print all primes between 1 and 1000
        for(int i = 1; i <= 1000; i++) {
            if(isPrime(i))
                System.out.print(i + " ");
        }

    }
}

```

```

/* this method takes an integer and returns true if n is
 * a prime integer and returns false otherwise.
 * Remember, a prime number has just two factors: 1 and itself */
public static boolean isPrime(int n) {

    if(n < 2) // no prime number less than 2
        return false;

    boolean ans = true; // assume n is prime

    //find factors of n
    for(int i = 2; i <= n/2; ++i)
        if(n % i == 0) {
            //i.e., i is a factor n. Therefore, n is not prime
            ans = false; // alter our initial assumption
            break;
        }

    return ans;
}
}

```

### Program 19

```

/*This program will print all the perfect numbers between 1 and 1000.
 * a perfect number is a positive integer greater than 1 whose sum of
 factors excluding number equals the number itself. E.g., 6 is perfect
 because 6 = 1 + 2 + 3.
 * The program uses the method isPefect to determine if a given integer
 * n is perfect or not.
 * */

```

```

//Filename: PerfectNumbers.java
public class PerfectNumbers {

    public static void main(String[] args) {

        for(int i = 1; i <= 1000; i++)
            if(isPerfect(i))
                System.out.print(i + " ");
    }

    public static boolean isPerfect(int n) {

        int sumOfFactors = 0;
        for(int i = 1; i < n; i++)
            if(n % i == 0) // i is a factor of n.
                sumOfFactors += i; //add i to the factors of n
    }
}

```

```

        return n == sumOfFactors;
    }
}

```

### Method 1

```

/* method that returns the largest of two elements
 * */
public static double max(double num1, double num2) {
    return num1 > num2 ? num1 : num2;
}

```

### Method 2

```

/*method that takes an array as argument computes and returns the
elements of the array */
public static double sumArrayElements(double[] array) {

    double sum = 0.0;
    for(double v: array)
        sum += v;

    return sum;
}

```

### Method 3

```

/*This method takes two arrays, A and B, and displays
 * those elements that are common to the two arrays.
 * */
public static void printCommonArrayElements(double[] A, double[] B)
{
    for(int i = 0; i < A.length; i++) {
        for(int j = 0; j < B.length; j++) {
            if(A[i] == B[j])
                System.out.print(A[i] + " ");
        }
    }
}

```

### Method 4

```

/* This method combines two arrays and
 * return a new array
 * */
public static double[] mergeTwoArrays(double[] A, double[] B) {

    int size = A.length + B.length;
    double[] mergedArray = new double[size];

    //copy the elements of array A into the mergedArray
    for(int i = 0; i < A.length; i++)

```

```
        mergedArray[i] = A[i];

        //copy the elements of array B into the mergedArray.
        for(int j = A.length, k = 0; j < mergedArray.length; j++, k++)
            mergedArray[j] = B[k];

        // return the mergedArray.
        return mergedArray;
    }
```

## ATTEMPT THE FOLLOWING PAST QUESTION PAPER

Kindly attempt the questions before looking at the solution provided after the question paper.



**YUSUF MAITAMA SULE UNIVERSITY, KANO**  
**FACULTY OF SCIENCE**  
**DEPARTMENT OF COMPUTER SCIENCE**  
**2020/2021 SECOND SEMESTER EXAMINATIONS – DEC 2022**  
**CSC2302 – COMPUTER PROGRAMMING II**

**Attempt Question 1 and any other 3 Questions**

**Time Allowed: 3 hours**

**1. [22 marks]**

- a. **[6 marks]** Write a Java program that: declares an array of 50 integers, set the value of each of the elements to the square of its index. Compute and display **[2 marks]**:
- [1 mark]** The sum of all the elements
  - [1 mark]** The average of the elements
  - [2 marks]** The percentages of the elements equals, below and above the computed average.
- b. **[12 marks]** Write a Java program that reads in a sentence of at least 100 characters and do the following **[2 marks]**:
- [2 marks]** Count and display the number of times a, e and o each appears in the sentence.
  - [2 marks]** Output the sentence in reverse separating the characters of the sentence by an underscore (\_).
  - [3 marks]** Capitalize all the characters at even positions (note: position is index+1).
  - [3 marks]** Break the sentence into words and display only the ones containing either an 'o' or 'u' in them, each on a separate line.
- c. **[4 marks]** Identify and explain the errors in the following code:

```
Public class ClassA{  
    public static void method1(){  
        double x = 1.8;  
        int y = x;  
        Method2();  
    }  
    public void method2(){  
        System.out.println('Ina kwana');  
        return 0;  
    }  
}
```

**2. [16 marks]**

- a. **[6 marks]** While inspecting a code written by a friend of yours, you suddenly realized that, it is extremely difficult to follow, talk less of maintenance. Write some comprehensive paragraphs for him and perhaps others, pointing out that, like any other form of development process, software development requires the following of some well-defined protocols/standards for an effective and sound outcome. The paragraphs should unambiguously explain the protocols software developers need to follow in getting rid of such difficult to follow and maintained spaghetti codes.
- a. **[4 marks]** With the aid of suitable examples, illustrate the working of a “for loop” and “for-each loop” control structures of Java.
- b. **[3 marks]** Define what methods/functions are. Are they at all important? Justify your answer.
- c. **[3 marks]** Distinguish between classes and objects.

**3. [16 marks]**

- a. **[5 marks]**
    - i. **[1 mark]** Define an array.
    - ii. **[1.5 marks]** Clearly explain the following statement:  
`int x[] = new int[10];`
    - iii. **[2.5 marks]** Consider the following statement:  
`int x[] = {1,2,4,5,6,7};`  
how can the elements of array x be copied into an array say y?
  - b. **[5 marks]** Write a java method with the following signature:  
**`void mathArithOperation (int opr1, int opr2, char op)`**.  
The method should perform arithmetic operation on **opr1** and **opr2** (operands) based on the value of **op** (operator) and print the result in the format "**opr1 op opr2 is value**". E.g. **mathArithOperation(2, 3, '+')** should output **2 + 3 is 5**.
  - c. **[4 marks]** With the aid of suitable examples both diagrammatically and in code snippets, differentiate between **while** and **do...while** loops.
  - d. **[2 marks]** Clearly differentiate between implicit and explicit typecasting.
- 4. [16 marks]**
- a. **[3 marks]** Write a program that reads in a sentence, compute and output the number of occurrences of all the letters, digits and whitespaces in it.
  - b. **[5 marks]** Explain the idea of structured programming delving more into its components.
  - c. **[5 marks]**  
Explain (with examples, where possible) the usage and importance of each of the following as relates to java:
    - i. Method signature
    - ii. **static** method
    - iii. Scanner class
    - iv. substring method of String class
    - v. split method of String class
  - d. **[3 marks]** Distinguish between Method Overriding and Method Overloading
- 5. [15 marks]**
- a. **[5 marks]** According to Wikipedia, "a palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward." Write a program that accepts a string and checks whether it's a palindrome or not. E.g. string "mallam" is a palindrome.
  - b. **[3 marks]** Define expression and its components.
  - c. **[2 marks]** Why is Java called a strongly typed language while JavaScript is not?
  - d. **[5 marks]** If you have the source code for a Java program, and you want to run it, you will need both a compiler and an interpreter. Clearly explain the role of both, and hence conclude that, Java is indeed a platform independent language.
- 6. [15 marks]**
- a. **[5 marks]** With the aid of a suitable example, illustrate the working of "switch" control structure of Java.
  - b. **[4 marks]** Briefly define the following:
    - i. Function
    - ii. Variable
    - iii. Literal
    - iv. OOP
  - c. **[2 marks]** Write a meaningful program of your choice.
  - d. **[5 marks]** Some Websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:
    - A password must have at least 8 characters.
    - A password can consists of only letters, digits and an underscore.
    - A password must contain at least 2 digits.





**YUSUF MAITAMA SULE UNIVERSITY, KANO**  
**FACULTY OF SCIENCE**  
**DEPARTMENT OF COMPUTER SCIENCE**  
**2020/2021 SECOND SEMESTER EXAMINATIONS – DEC 2022**  
**CSC2302 – COMPUTER PROGRAMMING II**

**SOLUTION**

**Attempt Question 1 and any other 3 Questions**

**Time Allowed: 3 hours**

**7. [22 marks]**

- d. **[6 marks]** Write a Java program that: declares an array of 50 integers, set the value of each of the elements to the square of its index. Compute and display **[2 marks]**:
- iv. **[1 mark]** The sum of all the elements
  - v. **[1 mark]** The average of the elements
  - vi. **[2 marks]** The percentages of the elements equals, below and above the computed average.

Answer:

```
Public class NewClass{
    public static void main(String ar[]){
        int [] numbers = new int[100];
        //initializing array elements to a random integer
        // less than or equals to its index
        numbers[0] = 0;
        for (int i = 1; i < numbers.length; i++){
            numbers[i] = i*i;
        }
        //getting the sum
        int sum = 0;
        for (int i = 1; i < numbers.length; i++){
            sum = sum + numbers[i] ;
        }
        double average = sum/100.0;
        int pBelow = 0, pEquals = 0, pAvohe = 0;
        for (int i = 1; i < numbers.length; i++){
            if(numbers[i] < average) {
                pBelow++;
            }
            else if(numbers[i] > average) {
                pAbove++;
            }
            else{
                pEquals++;
            }
        }
        System.out.println("The sum is "+sum);
        System.out.println("The average is " + average);
        System.out.println("The percentage below average is "+
pBelow);
```

```

        System.out.println("The percentage above average is "+
pAbove);
        System.out.println("The percentage equals average is "+
pEquals);
    }

```

- e. **[12 marks]** Write a Java program that reads in a sentence of at least 100 characters and do the following **[2 marks]**:
- v. **[2 marks]** Count and display the number of times a, e and o each appears in the sentence.
  - vi. **[2 marks]** Output the sentence in reverse separating the characters of the sentence by an underscore (\_).
  - vii. **[3 marks]** Capitalize all the characters at even positions (note: position is index+1).
  - viii. **[3 marks]** Break the sentence into words and display only the ones containing either an 'o' or 'u' in them, each on a separate line.

Answer:

```

import java.util.Scanner;
public class ExamClass{
    public static void main(String a[]){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a sentence of at least 100
characters:");
        String sentence = input.nextLine();
        if(sentence.length() < 100) return;
        int countA = 0, countE = 0, countU=0;
        //i. count of a, e and u
        for(int i = 0; i<sentence.length(); i++){
            if(sentence.charAt(i) == 'a')countA++;
            if(sentence.charAt(i) == 'e')countE++;
            if(sentence.charAt(i) == 'u')countU++;
        }

        System.out.println("a appears "+countA+" times");
        System.out.println("e appears "+countE+" times");
        System.out.println("u appears "+countU+" times");
        //ii. characters in reverse separated by underscore
        for(int i = sentence.length()-1; i>=0; i--){
            System.out.print(sentence.charAt(i) + "_");
        }
        System.out.println();
        //iii. capitalizing characters at even position
        for(int i = 0; i<sentence.length(); i++){
            if((i+1)%2==0){
                System.out.print(Character.toUpperCase(sentence.char
At(i)));
            }
            else{
                System.out.print(sentence.charAt(i));
            }
        }
        // iv. words containing 'a' or 'u'

```

```

        String sentWords []= sentence.split(" ");
        for(String aWord : sentWords){
            if(aWord.contains("a") || Word.contains("u")){
                System.out.println(aWord);
            }
        }
    }
}

```

- f. **[4 marks]** Identify and explain the errors in the following code:

```

Public class ClassA{
    public static void method1(){
        double x = 1.8;
        int y = x;
        Method2();
    }
    public void method2(){
        System.out.println('Ina kwana');
        return 0;
    }
}

```

Answer:

**Answer:**

**1 mark each.**

line 1, the p of the **public** access modifier should be in lowercase.

line 4, there is no explicit conversion from double to integer in java. Hence, the assignment is invalid.

Line 5, an instance method (**method2()**) cannot be invoked within a static method (**method1()**)

Line 8, strings are enclosed within a double quote, therefore, the string “ina kwana” should be double-quoted.

Line 9, method2() has nothing to return, the statement return 0 should be removed.

## 8. **[16 marks]**

- b. **[6 marks]** While inspecting a code written by a friend of yours, you suddenly realized that it is extremely difficult to follow, and talk less of maintenance. Write some comprehensive paragraphs for him and perhaps others, pointing out that, like any other form of development process, software development requires the following of some well-defined protocols/standards for an effective and sound outcome. The paragraphs should unambiguously explain the protocols software developers need to follow in getting rid of such difficult to follow and maintained spaghetti codes.

**Answer:**

Students are expected to explain in some (say 3) comprehensive paragraphs the fundamental principles of developing software such as: Do not Repeat Yourself (DRY), Keep It Simple and Short (KISS) and Avoiding We Enjoy Typing (WET) and many others.

- c. **[4 marks]** With the aid of suitable examples, illustrate the working of a “for loop” and “for-each loop” control structures of Java.

**Answer:**

All of the versions of for loop are repetition control structure, these structures are used for performing tasks repeatedly. They are: the normal for loop and the for-each loop.

For loop has the following structure:

```
for (initialization(s); condition ; update(s)) {  
    //instructions to be executed repeatedly till the condition  
    evaluates to false  
}
```

The initialization takes place, then the condition is tested, if it evaluates to true, the instructions within the body are executed. After the execution, then the update follows, and the condition is tested again. This process continues till the condition evaluates to false and the statements that follows the loop (if any) are executed.

e.g. the following code outputs numbers from 1 to 10:

```
for (int i = 1; i<=10; i++){  
    System.out.println(i);  
}
```

For-each loop is a kind of loop that can applied on collections (such as an array), it's used for performing same task on elements of a given collection, example:

```
int [] numbers = {1, 2, 3, 4};  
for(int number : numbers){//read: for each integer in the collection  
of integers  
    //do the following  
    System.out.print(number);  
}
```

d. **[3 marks]** Define what methods/functions are. Are they at all important? Justify your answer.

**Answer:**

Methods are named group of reusable instructions that can be invoked/called when needed. Methods are important, because complex programs are broken down into manageable pieces, these pieces are nothing but methods. The lead to saving of time, reuse of codes, reducing lines of codes and ease of maintenance of codes.

e. **[3 marks]** Distinguish between classes and objects.

**Answer:**

**Class:** is a blueprint that defines the features (fields) and behaviors (methods) in a particular kind of object. It can also be defined as a group of objects of the same type. One can create as many objects as possible from a given class. Example:

```
Public class Person{  
  
}
```

**Object:** An object is a particular instance of a class that have both states and behaviors. It can also be defined as a real-world entity that can be uniquely identified. Summarily, objects have identity, type, states and behaviors.

9. **[16 marks]**

e. **[5 marks]**

i. **[1 mark]** Define an array.

**Answer:** is a fixed-size sequential collection of elements of the same type.

- ii. [1.5 marks] Clearly explain the following statement:

```
int x[] = new int[10];
```

**Answer:**

int x[] creates an array reference variable x that can be used to refer to elements of type int

new int[10] creates an array of 10 int values

= sign link the array reference variable to the new created array

- iii. [2.5 marks] Consider the following statement:

```
int x[] = {1, 2, 4, 5, 6, 7};
```

how can the elements of array x be copied into an array say y?

**Answer:**

An assignment statement can not copy elements of array x in to array y, it can only copy the reference variable of array x to array y. We can use a loop to copy the elements of x and assign them to y as follows:

```
int x[] = {1, 2, 4, 5, 6, 7};
```

```
int y[] = new int[x.length];
```

```
for(int i=0; i<x.length; i++){
```

```
    y[i] = x[i];
```

```
}
```

- f. [5 marks] Write a java method with the following signature:

**void mathArithOperation (int opr1, int opr2, char op).**

The method should perform arithmetic operation on **opr1** and **opr2** (operands) based on the value of **op** (operator) and print the result in the format "**opr1 op opr2 is value**". E.g. **mathArithOperation(2, 3, '+')** should output **2 + 3 is 5**.

**Answer:**

```
public void mathArithOperation (int opr1, int opr2, char
op) {
    int result;
    switch(op) {
        case '+': result = opr1 + opr2; break;
        case '-': result = opr1 - opr2; break;
        case '*': result = opr1 * opr2; break;
        case '%': result = opr1 % opr2; break;
        case '/': result = opr1 / opr2; break;
        default: System.out.print("invalid operator");
    }
    return;
    System.out.println(opr1 + " " + op + " " + opr2 + " is "
+ result);
}
```

It can also be implemented using IF statement.

- g. [4 marks] With the aid of suitable examples both diagrammatically and in code snippets, differentiate between **while** and **do...while** loops.

**Answer:**

**Do while:**

```
do{
//statements to be executed till the condition evaluates to false
```

```
}while(condition);
```

**While:**

```
while(condition){  
    //statements to be executed repeatedly until the  
    //condition evaluates to false  
}
```

In the case of do...while, the enclosed statements will be executed at least ones, while in the case of while loop the statements will be executed zero or more times depending upon the outcome of the specified condition.

- h. **[2 marks]** Clearly differentiate between implicit and explicit typecasting.

**Answer:**

Typecasting refers to changing an entity of one datatype into another. Implicit typecasting is automatically done by the compiler, this is done in java only if the conversion will not temper with the actual semantic of the value in question, for example, there is automatic conversion from integers to real numbers. On the other, explicit typecasting has to be done manually (in code). This is necessary only if the conversion will temper with the actual semantic of the value, for example, converting a double number to an integer. 14.5 will either evaluates to 14 (lower bound) or 15 (upper bound). In both cases, the actual semantic of the value 14.5 has been tempered with.

**10. [16 marks]**

- e. **[3 marks]** Write a program that reads in a sentence, compute and output the number of occurrences of all the letters, digits and whitespaces in it.

**Answer:**

```
import java.util.Scanner;  
public class Question4a{  
    public static void main(String [] args){  
        Scanner input = new Scanner(System.in);  
        String theString;  
        System.out.print ("Enter a String:");  
        theString = input.nextLine();  
        int nLetters = 0, nDigits = 0, nWhitespace = 0;  
        for (int i = 0; i < theString.length(); i++){  
            char aChar = theString.charAt(i);  
            if (Character.isLetter(aChar)){  
                nLetters++;  
            }  
            else if (Character.isDigit(aChar)){  
                nDigits++;  
            }  
            else if (Character.isWhitespace(aChar)){  
                nWhitespace++;  
            }  
        }  
        System.out.println(nLetters + " letters");  
        System.out.println(nDigits + " digits");  
        System.out.println(nWhitespace + " whitespaces");  
    }  
}
```

- f. **[5 marks]** Explain the idea of structured programming delving more into its components.

**Answer:**

Students are expected to clearly explain what structured programming is and its components. These components are: functions, control structures and blocks.

- g. **[5 marks]**

Explain (with examples, where possible) the usage and importance of each of the following as relates to java:

- vi. Method signature
- vii. **static** method
- viii. Scanner class
- ix. substring method of String class
- x. split method of String class

**Answer:**

- i. Method signature: comprises of method name, parameter list and parameter types
- ii. **static** method: a method that is directly to a class itself not to any particular object of the class. in other words it can be invoked without creating object of the class.
- iii. Scanner class: : An object of Scanner class can be used to read data of any type from standard input (such as keyboard) or a file. This can be achieved using the methods such as: `nextInt()`, `nextDouble()`, etc.
- iv. substring method of String class: it is used for getting a substring of a particular string. There are two variants of this method: `String substring(int beginIndex):` returns the substring starting from the specified index (`beginIndex`) till the end of the string. For e.g. `"Hausa".substring(1)` would return `"ausa"`. The other variant is `String substring(int beginIndex, int endIndex):` returns string starting the given index (`beginIndex`) till the specified index (`endIndex`) excluding the character at the `endIndex`.
- v. split method of String class: is used for splitting a given string into multiple parts. Example: the following will split the string `s` into 3 parts with space as the splitting point and return the parts as an array of strings.

```
String s = "he is nice";
```

```
String [] parts = s.split(" ");
```

- h. **[3 marks]** Distinguish between Method Overriding and Method Overloading

**Answer**

A java class can contain two or more methods with the same name, provided that those methods accept different parameters, and this is termed method overloading. On the other hand, method overriding occurs when a subclass declares a method that has the same signature as a public method of the base class, in this case, the subclass version of the method overrides the base class version of the method.

## 11. [15 marks]

- e. **[5 marks]** According to Wikipedia, "a palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward." Write a program that accepts a string and checks whether it's a palindrome or not. E.g. string "mallam" is a palindrome.

**Answer:**

```
import java.util.Scanner;  
public class Question5a{
```

```

public static void main(String [] args){
    Scanner input = new Scanner(System.in);
    String theString;
    System.out.print ("Enter a String:");
    theString = input.nextLine();
    theString = theString.toLowerCase();
    int stringLength = theString.length();
    boolean isPalindrome = true;
    for (int i = 0, int j = stringLength - 1 ; i <
stringLength/2; i++, j--){
        char leftChar = theString.charAt(i);
        char rightChar = theString.charAt(j);
        if (leftChar != rightChar ){
            isPalindrome = false;
            break;
        }
    }
    if (isPalindrome){
        System.out.print ("The string is a palindrome");
    }
    else{
        System.out.print ("The string is not a palindrome");
    }
}

```

- f. **[3 marks]** Define expression and its components.

**Answer:**

Expression is a piece of program code that represents or computes a value. The components are variables, constants and function calls. Students are expected to briefly define the components.

- g. **[2 marks]** Why is Java called a strongly typed language while JavaScript is not?

**Answer:**

Java is strongly typed language because all variables must be declared with a data type, in other words, a variable of one type cannot hold the value of another different type.

- h. **[5 marks]** If you have the source code for a Java program, and you want to run it, you will need both a compiler and an interpreter. Clearly explain the role of both, and hence conclude that, Java is indeed a platform independent language.

**Answer:**

Compilation is the process of translating a program written in High-Level Language into a machine code for the purpose of creating an executable program. Compilation process is achieved using a Compiler. Interpretation is the process of executing a program written in a high-level language using an Interpreter. An interpreter translates high-level instructions into an intermediate form, which it then executes. Java is both interpreted and compiled programming language. Java programs are compiled into a virtual machine codes called Java Bytecode, which can then be interpreted/executed using Java Virtual Machine (JVM). There is JVM for windows, mac OS and Linux. This is one of the greatest features of Java language (cross-platform). Hence Java is a platform independent language.

Students can also explain JRE, JDK and all related terms.

## 12. [15 marks]



- a. **[5 marks]** With the aid of a suitable example, illustrate the working of “switch” control structure of Java.

**Answer:**

Switch is one of the selection control structures in JavaS. The syntax of a switch is as follows:

```
switch (expression){
    case 1: statement(s); break;
    case 2: statement(s); break;
    case 3: statement(s); break;
    ...
    case n: statement(s); break;
    default: statement(s);
}
```

This is how it works:

- i. The expression is evaluated once
- ii. The value of the expression is compared with the values of each case
- iii. If there is a match, the associated block of code is executed.

**Example:**

```
int x = 4;
switch (x){
    case 0: System.out.print("Today is Sunday");
            break;
    case 1: System.out.print ("Today is Monday");
            break;
    case 2: System.out.print ("Today is Tuesday");
            break;
    case 3: System.out.print ("Today is Wednesday");
            break;
    break;
    case 4: System.out.print ("Today is Thursday"); break;
    case 5: System.out.print ("Today is Friday"); break;
    case 6: System.out.print ("Today is Saturday"); break;
}
```

- b. **[4 marks]** Briefly define the following:

- v. Function
- vi. Variable
- vii. Literal
- viii. OOP

**Answer**

- i. Function: named group of reusable instructions and can be called when needed.
- ii. Variable: Symbol that represents storage location on computers memory
- iii. Literal: name for constant values
- iv. OOP: is a programming paradigm that is centered around the idea of objects.

- c. **[2 marks]** Write a meaningful program of your choice.

**Answer:**

Any valid and meaningful Java program

- d. **[5 marks]** Some Websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:

- A password must have at least 8 characters.
- A password can consists of only letters, digits and an underscore.
- A password must contain at least 2 digits.

**Answer:**

```
public static boolean isPasswordValid(String pswd){
    int digitsCountRule = 0;
    if(pswd.length() < 8){
        return false; //Invalid Password
    }
    for(int i=0; i<pswd.length(); i++){
        char aChar = pswd.charAt(i);
        if(!(Character.isDigit(aChar)||
Character.isLetter(aChar) || aChar == '_')){
            return false; //Invalid Password
        }
        if(Character.isDigit(aChar)){
            digitsCountRule += 1;
        }
    }
    if(digitsCountRule < 2){
        return false; //Invalid Password
    }
    return true; //The Password is Valid
}
```