ICSE Solved Paper 2009

Q1. (a) Why is a class called a *factory of objects*? [2 marks]

A1. (a) A class is called a factory of objects because it forms the blueprint from which multiple objects can be created. It is like a machine in a factory that keeps on producing objects. We can create as many objects of a class as we wish, with each object having its own set of instance variables. Each object of a class will have the same set of attributes and exhibit similar behaviour. Yet, due to the values of the data members, each is still distinct. Hence, a class is called a factory of objects.

Q1. (b) State the difference between a boolean literal and a character literal. [2 marks]

A1. (b)

|  |  |
| --- | --- |
| **BOOLEAN LITERAL** | **CHARACTER LITERAL** |
| Data type: boolean | Data type: char |
| Values: true or false only | Values: any char value |
| Value cannot be converted into an int | Value can be converted into an int |
| Space taken up depends on the implementation, and is not known | Space taken up is 2 bytes |

Q1. (c) What is the *use* and *syntax* of a ternary operator? [2 marks]

A1. (c) The ternary operator is a short-cut for a commonly used implementation of the if … else statement. It returns one value if the condition is true, and a second value if the condition is false.

The ternary operator is ?:

Syntax:

condition ? value-1 : value-2

Example:

The code

int x = 10;

if(x > 5)

 System.out.println(1);

else

 System.out.println(2);

can be replaced using the ternary operator with the code:

int x = 10;

System.out.println( (x > 5) ? 1 : 2 );

Q1. (d) Write one word answer for the following:

i) A method that converts a string to a primitive integer data type

ii) The default initial value of a boolean variable data type

[2 marks]

A1. (d)

i) Integer.parseInt()

ii) false

Q1. (e) State one similarity and one difference between *while* and *for* loop. [2 marks]

A1. (e)

Similarity: Both loops are entry-controlled loops (the do..while is exit-controlled, but here we are asked about while and for, not do..while).

Difference: The for portion has three sections, and all of them are optional. The sections can be omitted as long as the semi-colon is mentioned. In the while portion, there is only one section which is the condition, and it cannot be optional; the condition of a while loop must be specified.

Q2. (a) Write the function prototype for the function “sum” that takes an integer variable (x) as its argument and returns a value of float data type. [2 marks]

A2. (a) float sum(int x)

Q2. (b) What is the use of the keyword *this*? [2 marks]

A2. (b) The keyword this is used to indicate the current object. The keyword this followed by a period followed by the name of the instance variable can be used inside a member method to specifically indicate the instance variable of the calling object. This is particularly useful when an argument or some local variable in the same method has the same name as the instance variable.

The keyword this is also used as a function like this(…) inside a constructor to call an overloaded constructor of the same class. If used like this, it must be the first statement of the constructor.

Q2. (c) Why is a class known as a composite data type? [2 marks]

A2. (c) A class is known as a composite data type because it can have, as its instance variables, more than one instance variable, with each of these instance variables being either a primitive data type or a reference to another class’ object. For example, a class Car can have String model and int maxSpeed as its instance variables.

Q2. (d) Name the keyword that:

i) is used for allocating memory to an array

ii) causes the control to transfer back to the method call

A2. (d)

i) new

ii) return

Q2. (e) Differentiate between pure and impure functions. [2 marks]

A2. (e)

|  |  |
| --- | --- |
| **PURE FUNCTIONS** | **IMPURE FUNCTIONS** |
| Produces the same result (return value) for the same set of parameters, in each call | Produces different result for the same set of parameters, in each call |
| The execution of the function does not result in any side effect (such as the change in state of a mutable object, or an output to an I/O device.) | The execution of the function may result in an observable side effect. |
| Example: Math.sqrt() | Example: Math.random() – returns a random value. |
| Example: Math.sin() | Example: System.out.println() since it causes an output to an I/O device (monitor) as a side effect. |

Q3. (a) Write an expression for

(a + b)n

———

√3 + b

[2 marks]

A3. (a) Math.pow( a+b, n ) / ( Math.sqrt(3) + b )

Q3. (b) The following is a segment of a program.

x = 1; y = 1;

if(n > 0)

{

 x = x + 1;

 y = y - 1;

}

What will be the value of x and y, if n assumes a value (i) 1 (ii) 0?

[2 marks]

A3. (b)

i) x is 2, y is 0

ii) x is 1, y is 1

Q3. (c) Analyze the following program segment and determine how many times the body of loop will be executed (show the working).

x = 5; y = 50;

while(x<=y)

{

 y=y/x;

 System.out.println(y);

}

[2 marks]

A3. (c) The working is shown in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **X** | **Y** | **CONDITION X<=Y** | **NEW Y** | **OUTPUT** | **REMARKS** |
| 5 | 50 | true | 10 | 10 | Iteration 1 |
| 5 | 10 | true | 2 | 2 | Iteration 2 |
| 5 | 2 | false |  |  | No 3rd iteration |

The body of the loop will be executed 2 times. The output will be:



Q3. (d) When there are multiple definitions with the same function name, what makes them different from each other? [2 marks]

A3. (d) The concept is called function overloading. Here, the signature of the function makes the functions different from each other. The signature of a function includes the name of the function, and the number and data types of the arguments to the function. For example, the prototypes f(), f(int x), and f(int x, int y), and f(String x) are all overloaded functions. However, the just by changing the name of the actual parameter, the signature does not change; for example: f(int x) and f(int y) are not allowed, since both are f(int).

Q3. (e) Given that int x[][] = { {2,4,6}, {3,5,7} };

What will be the value of x[1][0] and x[0][2] ?

[2 marks]

A3. (e)

x[1][0] = 3

x[0][2] = 6

Q3. (f) Give the output of the following code segment when (i) opn = ‘b’ (ii) opn = ‘x’ (iii) opn = ‘a’.

switch(opn)

{

 case 'a':

 System.out.println("Platform Independent");

 break;

 case 'b':

 System.out.println("Object Oriented");

 case 'c':

 System.out.println("Robust and Secure");

 break;

 default:

 System.out.println("Wrong Input");

}

[3 marks]

A3. (f)



Q3. (g) Consider the following code and answer the questions that follow:

class academic

{

 int x, y;

 void access()

 {

 int a, b;

 academic student = new academic();

 System.out.println("Object created");

 }

}

i) What is the object name of class academic?

ii) Name the class variables used in the program?

iii) Write the local variables used in the program.

iv) Give the type of function used and its name.

[4 marks]

A3. (g)

i) student

ii) there are no class variables (static). The instance variables used are x and y.

iii) a and b

iv) Name: access()

Type: member function (not static)

Return value: none (void)

Q3 (h) Convert the following segment into an equivalent *do* loop.

int x,c;

for(x=10,c=20;c>10;c=c-2)

x++;

[3 marks]

A3. (h)

int x,c;

x = 10;

c = 20;

do

{

 x++;

 c=c-2;

}while(c>10);

Note: At the end, the value of x = 15, and c = 10

Q4. An electronics shop has announced the following seasonal discounts on the purchase of certain items.

|  |  |  |
| --- | --- | --- |
| **PURCHASE AMOUNTIN RS.** | **DISCOUNT ONLAPTOP** | **DISCOUNT ONDESKTOP PC** |
| 0 – 25000 | 0.0% | 5.0% |
| 25001 – 57000 | 5.0% | 7.6% |
| 57001 – 100000 | 7.5% | 10.0% |
| More than 100000 | 10.0% | 15.0% |

Write a program based on the above criteria to input name, address, amount of purchase and type of purchase (L for Laptop and D for Desktop) by a customer. Compute and print the net amount to be paid by a customer along with his name and address.

(Hint: discount = (discount rate/100)\* amount of purchase

Net amount = amount of purchase – discount)

[15 marks]

A4.

import java.util.\*;

class Q4

{

 public static void main(String[] args)

 {

 // ask user information

 Scanner sc = new Scanner( System.in );

 String name, address;

 int amount;

 char type = ' '; // space

 String temp;

 System.out.print("Enter name: ");

 name = sc.nextLine();

 System.out.print("Enter address: ");

 address = sc.nextLine();

 System.out.print("Enter amount of purchase: ");

 amount = sc.nextInt();

 temp = sc.nextLine(); // for trailing new line

 // ask the type, if incorrect ask again

 do

 {

 System.out.print("Enter L for Laptop, D for Desktop: ");

 type = sc.nextLine().toUpperCase().charAt(0);

 if( "LD".indexOf(type) < 0 )

 System.out.println("Incorrect type. Try again...");

 }while("LD".indexOf(type) == -1);

 // compute discount etc

 double discountRate = 0.0, discount = 0.0;

 double netAmount = 0.0;

 switch(type)

 {

 case 'L':

 if(amount <= 25000)

 discountRate = 0.0;

 else if( (amount >= 25001) && (amount <= 57000) )

 discountRate = 5.0;

 else if( (amount >= 57001) && (amount <= 100000) )

 discountRate = 7.5;

 else

 discountRate = 10.0;

 break;

 case 'D':

 if(amount <= 25000)

 discountRate = 5.0;

 else if( (amount >= 25001) && (amount <= 57000) )

 discountRate = 7.6;

 else if( (amount >= 57001) && (amount <= 100000) )

 discountRate = 10.0;

 else

 discountRate = 15.0;

 break;

 }

 discount = discountRate / 100.0D \* amount;

 // round to two places of decimal

 discount = Math.round(discount \* 100.0);

 discount = discount / 100;

 netAmount = amount - discount;

 // display the output

 System.out.println("BILL");

 System.out.println("Customer name : " + name);

 System.out.println("Address : " + address);

 System.out.println("Purchase amount: " + amount);

 System.out.println("Purchase type : " + type);

 System.out.println("Discount % : " + discountRate);

 System.out.println("Discount Amount: " + discount);

 System.out.println("Net amount : " + netAmount);

 }

}

2 Sample outputs are shown below:



Q5. Write a program to generate a triangle or an inverted triangle till *n* terms based upon the user’s choice of triangle to be displayed.

|  |  |
| --- | --- |
| **EXAMPLE 1** | **EXAMPLE 2** |
| Input: Type 1 for a triangle andtype 2 for an inverted triangle1Enter the number of terms5Output:12 23 3 34 4 4 45 5 5 5 5 | Input: Type 1 for a triangle andtype 2 for an inverted triangle2Enter the number of terms6Output:6 6 6 6 6 65 5 5 5 54 4 4 43 3 32 21 |

[15 marks]

A5.

import java.util.\*;

class Q5

{

 public static void main(String[] args)

 {

 int choice = -1;

 int n = 0;

 do

 {

 choice = askChoice();

 switch(choice)

 {

 case 1:

 n = askInt("Enter the number of terms: ");

 doTriangle(n);

 break;

 case 2:

 n = askInt("Enter the number of terms: ");

 doInvertedTriangle(n);

 break;

 case 0:

 System.out.println("Exiting program. Thank you!");

 break;

 }

 }while(choice != 0);

 }

 // Method to ask 1 for triangle, 2 for inverted triangle

 // 0 for exit

 // if wrong entry, ask again

 public static int askChoice()

 {

 int x = -1;

 do

 {

 x = askInt("Type 1 for a triangle and type 2 for an inverted triangle, 0 for exit: ");

 if( (x < 0) || (x > 2) )

 System.out.println("Wrong input. Try again...");

 }while( !( (x >= 0) && (x <= 2) ) );

 return x;

 }

 // Utility method

 // Ask user to enter an integer

 // The prompt is passed as a parameter

 public static int askInt(String prompt)

 {

 Scanner sc = new Scanner( System.in );

 int n;

 String temp;

 System.out.print( prompt );

 n = sc.nextInt();

 temp = sc.nextLine(); // for the trailing newline

 return n;

 }

 // triangle

 public static void doTriangle(int n)

 {

 int i, j;

 for(i=1; i <= n; i++)

 {

 for(j=1; j <= i; j++)

 {

 System.out.print(i + " ");

 }

 System.out.println();

 }

 }

 // inverted triangle

 public static void doInvertedTriangle(int n)

 {

 int i, j;

 for(i=n; i>0; i--)

 {

 for(j=1; j<=i; j++)

 {

 System.out.print(i + " ");

 }

 System.out.println();

 }

 }

}

Sample output:



Q6. Write a program to input a sentence and print the number of characters found in the longest word of the given sentence.

For example is S = “India is my country” then the output should be 7.

[15 marks]

A6.

import java.util.\*;

class Q6

{

 public static void main(String[] args)

 {

 // Ask user to enter a sentence.

 Scanner sc = new Scanner( System.in );

 System.out.print("Enter a sentence: ");

 String sentence = sc.nextLine();

 // Break the sentence into words.

 // Use an overloaded parameterized constructor to use multiple

 // word separators.

 String[] words = new String[sentence.length()];

 int wordCount = 0;

 StringTokenizer st = new StringTokenizer(sentence, " ,!.?;()");

 while(st.hasMoreTokens())

 {

 words[wordCount++] = st.nextToken();

 }

 // Compute the length of the longest word.

 int longestLength = 0;

 int i;

 for(i=0; i < wordCount; i++)

 if(words[i].length() > longestLength)

 longestLength = words[i].length();

 // Display the length of the longest word(s).

 System.out.println( "Length of the longest word(s): " + longestLength);

 }

}

Some sample outputs:



Q7. Write a class to overload a function num\_calc() as follows:

i) void num\_calc(int num, char ch) with one integer argument and one character argument, computes the square of integer argument if choice ch is ‘s’ otherwise finds its cube.

ii) void num\_calc(int a, int b, char ch) with two integer arguments and one character argument. It computes the product of integer arguments if ch is ‘p’ else adds the integers.

iii) void num\_calc(String s1, String s2) with two string arguments, which prints whether the strings are equal or not.

[15 marks]

A7.

class Q7

{

 // overloaded function #1

 void num\_calc(int num, char ch)

 {

 if(ch == 's')

 System.out.println("Square of " + num + ": " + (num\*num));

 else

 System.out.println("Cube of " + num + ": " + (num\*num\*num));

 }

 // overloaded function #2

 void num\_calc(int a, int b, char ch)

 {

 if(ch == 'p')

 System.out.println("Product of " + a + " and " + b + ": " + (a\*b));

 else

 System.out.println("Sum of " + a + " and " + b + ": " + (a+b));

 }

 // overloaded function #3

 void num\_calc(String s1, String s2)

 {

 if(s1.equals(s2))

 System.out.println(s1 + " and " + s2 + " are equal.");

 else

 System.out.println(s1 + " and " + s2 + " are not equal.");

 }

 // entry point main method

 public static void main(String[] args)

 {

 Q7 obj = new Q7();

 // calling overloaded function #1

 obj.num\_calc(3, 's');

 obj.num\_calc(4, 'c');

 // calling overloaded function #2

 obj.num\_calc(3, 4, 'p');

 obj.num\_calc(3, 4, 's');

 // calling overloaded function #3

 obj.num\_calc("HELLO", "HELLO");

 obj.num\_calc("MELLOW", "YELLOW");

 }

}

Output:



Q8. Write a menu driven program to access a number from the user and check whether it is a BUZZ number *or* to accept any two numbers and to print the GCD of them.

a) A BUZZ number is the number which either ends with 7 is is divisible by 7.

b) GCD (Greatest Common Divisor) of two integers is calculated by continued division method. Divide the larger number by the smaller; the remainder then divides the previous divisor. The process is repeated till the remainder is zero. The divisor then results the GCD.

[15 marks]

A8.

import java.util.\*;

class Q8

{

 public static void main(String[] args)

 {

 char menuOption = ' '; // space

 do

 {

 menuOption = askMenuOption();

 switch(menuOption)

 {

 case 'B':

 doBUZZ();

 break;

 case 'G':

 doGCD();

 break;

 case 'X':

 System.out.println("Exiting program. Thank you!");

 break;

 }

 }while( menuOption != 'X' ); // exit when X is entered

 }

 // ask menu option and return it

 // validate that the entry is B, G, or X, else ask again

 public static char askMenuOption()

 {

 Scanner sc = new Scanner( System.in );

 char option = ' '; // space

 do

 {

 System.out.print("Enter B for BUZZ, G for GCD, X to Exit: ");

 option = sc.nextLine().toUpperCase().charAt(0);

 if( "BGX".indexOf(option) < 0 )

 System.out.println("Incorrect option. Try again...");

 }while( "BGX".indexOf(option) < 0 );

 return option;

 }

 // BUZZ number logic

 public static void doBUZZ()

 {

 int n = askInt("Enter a number for BUZZ checking: ");

 if( ((n%10) == 7) || ((n%7)==0) )

 System.out.println(n + " is a BUZZ number.");

 else

 System.out.println(n + " is not a BUZZ number.");

 }

 // GCD logic

 public static void doGCD()

 {

 int n1 = askInt("Enter number 1 for GCD: ");

 int n2 = askInt("Enter number 2 for GCD: ");

 int nr = (n1 > n2) ? n1 : n2; // larger number

 int dr = (n1 > n2) ? n2 : n1; // smaller number

 while( (nr%dr) != 0 )

 {

 int remainder = nr % dr;

 nr = dr;

 dr = remainder;

 }

 System.out.println("The GCD of " + n1 + " and " + n2 + " is " + dr);

 }

 // utility method for prompting user

 // and returning an int entered by the user

 public static int askInt(String prompt)

 {

 Scanner sc = new Scanner( System.in );

 System.out.print(prompt);

 int n = sc.nextInt();

 String temp = sc.nextLine(); // for the trailing newline

 return n;

 }

}

Sample output:



Q9. The annual examination results of 50 students in a class is tabulated as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| Roll no. | Subject A | Subject B | Subject C |
| ……… | ……… | ……… | ……… |

Write a program to read the data, calculate and display the following:

a) Average mark obtained by each student.

b) Print the roll number and average marks of the students whose average mark is above 80.

c) Print the roll number and average marks of the students whose average mark is below 40.

[15 marks]

A9.

import java.util.\*;

class Q9

{

 public static void main(String[] args)

 {

 // ask user to enter the data

 Scanner sc = new Scanner( System.in );

 int N = 10; // use 50 in the exam

 int[] rollNo = new int[N];

 int[][] marks = new int[N][3];

 double[] average = new double[N];

 int i, j;

 String temp;

 for(i=0; i < N; i++)

 {

 System.out.print("Enter roll no and 3 marks for student# " + (i+1) + ": ");

 rollNo[i] = sc.nextInt();

 for(j=0; j < marks[i].length; j++)

 marks[i][j] = sc.nextInt();

 temp = sc.nextLine(); // for the trailing newline

 }

 // compute the average

 int sum;

 for(i=0; i < N; i++)

 {

 sum = 0;

 for(j=0; j < marks[i].length; j++)

 sum += marks[i][j];

 average[i] = (double)sum / (double)marks[i].length;

 }

 displayAverageOfEach(rollNo, average);

 displayAverageOf80Plus(rollNo, average);

 displayAverageOf40Minus(rollNo, average);

 }

 public static void displayAverageOfEach(int[] rollNo, double[] average)

 {

 int i;

 System.out.println("Average of each student");

 System.out.println("Roll no. Average");

 System.out.println("-------- -------");

 for(i=0; i < rollNo.length; i++)

 {

 printInt(rollNo[i], 8+1);

 printDouble(average[i], 7);

 System.out.println();

 }

 }

 public static void displayAverageOf80Plus(int[] rollNo, double[] average)

 {

 int i;

 System.out.println("Averages of students with more than 80");

 System.out.println("Roll no. Average");

 System.out.println("-------- -------");

 for(i=0; i < rollNo.length; i++)

 {

 if(average[i] > 80)

 {

 printInt(rollNo[i], 8+1);

 printDouble(average[i], 7);

 System.out.println();

 }

 }

 }

 public static void displayAverageOf40Minus(int[] rollNo, double[] average)

 {

 int i;

 System.out.println("Averages of students with less than 40");

 System.out.println("Roll no. Average");

 System.out.println("-------- -------");

 for(i=0; i < rollNo.length; i++)

 {

 if(average[i] < 40)

 {

 printInt(rollNo[i], 8+1);

 printDouble(average[i], 7);

 System.out.println();

 }

 }

 }

 public static void printInt(int n, int width)

 {

 String s = "" + n;

 printStr(s, width);

 }

 public static void printDouble(double d, int width)

 {

 int i = (int)Math.round(d \* 100.0);

 String D = "" + i;

 String s = D.substring(0, D.length()-2) + "." + D.substring( D.length() - 2 );

 printStr(s, width);

 }

 public static void printStr(String s, int width)

 {

 System.out.print(s);

 int spaces = width - s.length();

 while(spaces > 0)

 {

 System.out.print(" "); // single space

 spaces--;

 }

 }

}

Sample output with 10 students data:



