Information and Computer Science Department

First Semester 161

ICS 103 – Computer Programming in C

Major 01 Examination Key

Saturday, November 5, 2016 (9:00-11:00AM)

Duration: 120 minutes

|  |  |  |  |
| --- | --- | --- | --- |
| Name: |  | Lect Serial # |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID#: |  |  |  |  |  |  |  |  |  |

Please tick your section:

|  |  |
| --- | --- |
| Instructor | Section |
| Mr. Said Abdallah Muhammad | [ ] 01 (UT 7:00 – 7:50) [ ] 02 (UT 7:00 – 7:50) |
| Dr. Hamood Al-Jamaan | [ ] 03 (UT 8:00 – 8:50) [ ] 04 (UT 8:00 – 8:50) |
| Dr. Rafiul Hassan | [ ] 05 (UT 11:00 – 11:50) [ ] 06 (UT 11:00 – 11:50)  [ ] 07 (UT 13:10 – 14:00) [ ] 08 (UT 13:10 – 14:00) |
| Dr. Muhammad Balah | [ ] 09 (MW 7:00 – 7:50) [ ] 10 (MW 7:00 – 7:50) |
| Mr. Muhammad Aslam | [ ] 11 (MW 8:00 – 8:50) [ ] 12 (MW 8:00 – 8:50) |
| Dr. Samer Arafat | [ ] 13 (MW 9:00 – 9:50) [ ] 14 (MW 9:00 – 9:50)  [ ] 15 (MW 10:00 – 10:50) [ ] 16 (MW 10:00 – 10:50) |
| Dr. Louai Al-Awami | [ ] 17 (MW 11:00 – 11:50) [ ] 18 (MW 11:00 – 11:50)  [ ] 19 (MW 13:10 – 14:00) [ ] 20 (MW 13:10 – 14:00) |

Instructions**:**

1. Answer all questions. Make sure your answers are clear and readable.
2. Make sure there are **5** questions in **10** pages.
3. The exam is closed book and closed notes. No calculators or any helping aides are allowed. Make sure

to turn off your mobile phone and keep it in your pocket.

1. If there is no space on the front of a question’s page, use the back of the page. Indicate this clearly.

|  |  |  |  |
| --- | --- | --- | --- |
| Question # | Max Grade | Obtained Grade | Remarks |
| 1 | 15 |  |  |
| 2 | 8 |  |  |
| 3 | 9 |  |  |
| 4 | 10 |  |  |
| 5 | 9 |  |  |
| 6 | 13 |  |  |
| 7 | 16 |  |  |
| 8 | 20 |  |  |
| **Total** | **100** |  |  |

## Question 1 (15 points): (**1.5 pt each**)

Fill this table by selecting the correct solution for each of the following 10 questions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Answer | **B** | **B** | **A** | **C** | **D** | **C** | **A** | **A** | **C** | **A** |

1. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ converts an object file into an executable file.
   1. Compiler
   2. **Linker**
   3. Loader
   4. Word processor
2. C Language is an example of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. Machine language
   2. **High-Level language**
   3. Assembly language
   4. None of the above
3. Which of the following is used during the design step of software development?
   1. **Pseudocode**
   2. Integrated Development Environment (IDE)
   3. C compiler
   4. None of the above
4. If we define **int x=253;** and double **y=953.2;** which **printf** statement will result in the following output?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **9** | **5** | **3** | **.** | **2** |  |  |  |  | **2** | **5** | **3** |  |

* 1. printf("%3.1f%4d\n", y, x);
  2. printf("%6.1f%3d\n", y, x);
  3. **printf("%8.1f%7d\n", y, x);**
  4. printf("%6.5f%4d\n", y, x);

1. Given the following code fragment,

double temp = 27;

char temp\_unit= 'C';

show\_temperature(temp, temp\_unit);

Which prototype is correct for the function **show\_temperature** ?

* 1. int show\_temperature(double, char)
  2. void show\_temperature(double, char)
  3. int show\_temperature(double, char);
  4. **void show\_temperature(double, char);**

1. Consider the following code fragment. Which of the following expression should be used in **line 4** to have the correct program behavior?

|  |  |
| --- | --- |
| 1. | int x; |
| 2. | printf("Enter an integer number: "); |
| 3. | scanf("%d", &x); |
| 4. |  |
| 5. | if(odd) |
| 6. | printf("x is odd"); |
| 7. | else |
| 8. | printf("x is even"); |

* 1. odd = (x%2 != 1);
  2. odd = (x%2 !== 1);
  3. **odd = (x%2);**
  4. odd = (x/2);

1. Which of the following expressions is equivalent to the expression?

!((x || !y) && z)

* 1. **!x && y || !z**
  2. !(x && y) || !z
  3. (!x || y) && !z
  4. (x && !y) || z

1. Assuming a=4, b=6, c=0, the value of the following expression is:

(a > b >= c)

* 1. **1**
  2. 0
  3. Undefined
  4. Error

1. Suppose **displaySum** is a function with return type **void** and which displays the sum of two integers. Which of the following is a valid call to the function? Assume that x and y are initialized integer variables.
   1. void displaySum(int x, int y);
   2. displaySum(int x = 2, int y = 6);
   3. **displaySum(x, y);**
   4. printf("%d", displaySum(x, y));
2. Which of the following C expressions will yield the same result as 5 % 2 ?
   1. **5 - 5 / 2 \* 2**
   2. 5 - 2 / 5 \* 5
   3. 2 - 2 / 5 \* 5
   4. 2 - 5 / 2 \* 2

Question 2 (8 points): Write the equivalent C condition so that each of the following statements is satisfied.

|  |  |
| --- | --- |
| **Statement** | **Equivalent C condition** |
| m is an odd integer or y is an even integer | **m % 2 != 0 || y % 2 == 0** |
| t and w are not more than 5 | **t <= 5 && w <= 5** |
| x is outside the interval [7,25] | **x < 7 || x > 25** |
| b takes all values between 10 and 5 inclusive | **b > = 5 && b <= 10** |

Question 3 (9 points): Find the values of the following C expressions.

|  |  |
| --- | --- |
| **Expression** | **Value 1.5 pt for each correct value** |
| 127 % 20 / 5 | **1** |
| 3 + 3 / 2.0 | **4.5** |
| 5 > 3 > 1 | **0 accept false** |
| 5 - 5 != 5 - 3 >= 2 | **1 accept true** |
| (int)9.9 / (double)2 | **4.5** |
| (double)(7 / 2) | **3.0** |

Question 4 (10 points): What will be the output of the following code fragments?

|  |  |
| --- | --- |
| **Code** | **Output** |
| 1 pt    int i = 10,j = 12,k = 5;  i = j **==** k;  printf("%d",i); | **0** |
| 2 pts  if(3 != 10)  printf("A100");  else  printf(" B100");  printf(" C100"); | **A100 C100** |
| 3 pts  int A = 20, B = 15, C = 18;  if(A >= B)  if(B >= C)  if(A > C)  printf("AA\n");  else  printf("BB\n");  else  printf("CC\n"); | **CC** |
| 4 pts  int X = 0;  int Y = 1;  if (X)  printf("X\n");  if (Y)  printf("Y\n");  if (X = Y)  printf("X=Y\n");  else  printf("X!=Y\n"); | **Y**  **X=Y** |

**Question 5 (9 points):** Assume that you wrote a program asking the user to enter an integer number in the range [4, 9]. The number typed by the user is stored in an **integer** variable called x. Consider the following if-else-if statement

if(x **==** 4 || x **==** 6 || x **==** 8)

printf("even");

else if (x **==** 5 || x **==** 7)

printf ("prime");

else if (x **==** 9)

printf("odd");

else

printf("Wrong input");

Now instead of using **if-else-if** you are asked to rewrite the above code using **switch** statement. The two codes must do the same job.

**switch (x) {**

**case 4:**

**case 6:**

**case 8:printf("even");**

**break;**

**case 5:**

**case 7:printf ("prime");**

**break;**

**case 9:printf("odd");**

**break;**

**default: printf("wrong input");**

**}**

Alternatively:

**switch (x) {**

**case 4: printf("even");**

**break;**

**case 6: printf("even");**

**break;**

**case 8: printf("even");**

**break;**

**case 5: printf ("prime");**

**break;**

**case 7: printf ("prime");**

**break;**

**case 9: printf("odd");**

**break;**

**default: printf("wrong input");**

**}**

**Question 6 (13 points):** Write a complete C program that prompts the user to input a value of temperature in either Fahrenheit (F) or Celsius (C). If the value entered is in Fahrenheit (F) (by typing F or f after the value), the program converts it to Celsius (C). Likewise, if the input temperature is in Celsius (C) (by typing C or c after the value), the program converts it to Fahrenheit (F). The result is then displayed on the screen. The program must detect the unit of the input value automatically then carries out the conversion. If the input unit is not F or C, lower or uppercase, the program displays "invalid input " message.

Note: where : Temperature in Celsius, : Temperature in Fahrenheit.

Three sample program runs:

|  |
| --- |
| **Enter the temperature (in C or F):**  **30.7 C**  **The temperature you entered equals 87.26 F** |
| **Enter the temperature (in C or F):** **90 f**  **The temperature you entered equals 32.22 C** |
| **Enter the temperature (in C or F):** **87.2 T**  **Invalid input** |

**#include <stdio.h>**

**int main(void) {**

**double source\_temp, target\_temp;**

**char temp\_type;**

**printf("Enter the temperature (in C or F): ");**

**scanf("%lf %c",&source\_temp,&temp\_type);**

**if(temp\_type =='f' || temp\_type =='F'){**

**target\_temp = (source\_temp - 32)\*5/9;**

**printf("The temperature you entered equals %.2f C",target\_temp);**

**}**

**else if(temp\_type =='c' || temp\_type =='C'){**

**target\_temp = source\_temp \*9/5 + 32;**

**printf("The temperature you entered equals %.2f F",target\_temp);**

**}**

**else{**

**printf("Invalid Input");**

**}**

**return 0;**

**}**

**Question 7 (16 points):** Write a C program that prompts the user and reads a 3-digit positive integer number. It then finds the ones, tens, and hundreds and display them on the screen as shown in the sample program runs. If the input is not a 3-digit positive integer number, your program displays an error message.

Two sample program runs:

|  |
| --- |
| Enter a positive 3 digit number: -67 Error: Your number must be between 100 and 999 |
| Enter a positive 3 digit number: 583 Hundreds = 5  Tens = 8  Ones = 3 |

#**include <stdio.h>**

**int main() {**

**int n,ones,tens,hundreds,rem;**

**printf("Enter a positive 3 digit number: ");**

**scanf("%d",&n);**

**if(n >= 100 && n <= 999){**

**hundreds = n / 100;**

**rem = n % 100;**

**tens = rem / 10;**

**ones = rem % 10;**

**printf("Hundreds= %d\nTens= %d\nOnes= %d\n",hundreds,tens,ones);**

**}**

**else**

**printf("Your number must be between 100 and 999\n");**

**return 0 ;**

**}**

**Question 8 (20 points):** An employee is paid according to the formulas shown below.

Write a complete C program that must contain, in addition to the main functions, two functions **getGrossSalary** and **getNetSalary**.

**getGrossSalary** receives the hours and the wage and returns the gross salary.

**getNetSalary** receives the gross salary and returns the net salary.

In the main function, you read from the user the number of hours and the wage. If number of hours or wage is zero or negative, the main function displays an error message; otherwise it calls the functions **getGrossSalary and getNetSalary**. After calling **getGrossSalary** and **getNetSalary** functions, your program prints the result as shown in the sample program run.

**Note:**

* You must put the definition of the **main** function before those of **getGrossSalary** and **getNetSalary**.
* The functions **getGrossSalary** and **getNetSalary** must not contain any **printf** or **scanf** calls. All reading and printing must be done in the **main** function.

Two sample program runs:

|  |  |
| --- | --- |
| Enter hours and wage in Saudi Riyals: 50 60.80  Gross salary = 3344.00 Saudi Riyals  Net salary = 3226.80 Saudi Riyals | Enter hours and wage in Saudi Riyals: -34 50.7  Wrong input |

**#include <stdio.h>**

**double getGrossSalary(int hours, double wage);**

**double getNetSalary(double grossSalary);**

**int main(void) {**

**int hours;**

**double wage, grossSalary,netSalary;**

**printf("Enter hours and wage in Saudi Riyals: ");**

**scanf("%d%lf",&hours,&wage);**

**if(hours <= 0 || wage <= 0)**

**printf("Error: Invalid input\n");**

**else{**

**grossSalary = getGrossSalary(hours,wage);**

**netSalary = getNetSalary(grossSalary);**

**printf("Gross salary = %.2f Saudi Riyals\n", grossSalary);**

**printf("Net Salary = %.2f Saudi Riyals\n", netSalary);**

**}**

**return 0 ;**

**}**

**double getGrossSalary(int hours, double wage){**

**double salary;**

**if(hours <= 40)**

**salary = 40 \* wage;**

**else**

**salary = 1.5 \* wage \* (hours - 40) + 40 \* wage;**

**return salary;**

**}**

**double getNetSalary(double grossSalary){**

**double netSalary;**

**if(grossSalary < 1000)**

**netSalary = grossSalary;**

**else**

**netSalary = grossSalary + 5 - 0.05 \* grossSalary;**

## **return netSalary;**

## **}**