11th First Revision Answer Key 2018-19
Computer Science

1-Marks

<table>
<thead>
<tr>
<th>Q.No</th>
<th>Ans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>C</td>
</tr>
<tr>
<td>10</td>
<td>B</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>B</td>
</tr>
<tr>
<td>13</td>
<td>B</td>
</tr>
<tr>
<td>14</td>
<td>B</td>
</tr>
<tr>
<td>15</td>
<td>B</td>
</tr>
</tbody>
</table>

2-Marks

16. **Input Unit**
- Input unit is used to feed any form of data to the computer, which can be stored in the memory unit for further processing.
- Example: Keyboard, mouse, etc.

**Output Unit**
- An Output Unit is any hardware component that conveys information to users in an understandable form.
- Example: Monitor, Printer etc.

17. This is a one of the features of Operating System. It has two or more processors for a single running process (job). Processing takes place in parallel is known as parallel processing. Each processor works on different parts of the same task or on two or more different tasks. Since the execution takes place in parallel, this feature is used for high speed execution which increases the power of computing.

18. **Save**
- Save Option is used to Store the Data / File into a computer.
- To Save the file Choose File → Save Option or Ctrl+S.

**Save As**
- Save As Option is uses to Store the Data / File into a computer as in Another name (or) place as like save.
- To Save As the file Choose File → Save As Option.

19. High-Definition Multimedia Interface is an audio/video interface which transfers the uncompressed video and audio data from a video controller, to a compatible computer monitor, LCD projector, digital television etc.
20. int main ()
{
    int i;
    for(i = 21; i < 31; i++)
        cout << "value of i : " << i << endl;
    return 0;
}

21. Polymorphism is the ability of a message or function to be displayed in more than one form.

22. Structure:
   Structure is a user-defined which has the combination of data items with different data types. This allows to group of variables of mixed data types together into a single unit.
   Array:
   An array is a collection of variables of the same type that are referenced by a common name.

23. Array:
   The ability of the function to process the message or data in more than one form is called as function overloading. In other words function overloading means two or more functions in the same scope share the same name but their parameters are different.

3-Marks

25.

i) Add : \( 1101010_2 + 101101_2 \)
   
   01101010
   (+) 00101101
   
   --------------
   10010111

Prepared By : T.Thirumalai M.Sc(CS),B.Ed,
ii) Add: \((-22)_{10} + 15_{10}\)

22 in Binary Digit \(\rightarrow 0001 0110\)  
15 \(\rightarrow 0000 1111\)

1's Complement \(\rightarrow 1110 1001\)

2's Complement \(\rightarrow +1\)

\[\begin{array}{c}
\hline
& 0001 0110 & + & 0000 1111 \\
\hline
\text{Ans: } & 1111 1010 (-22) \\
\hline
\end{array}\]

\((-6) \rightarrow 1111 1001\)

A mobile operating system controls a mobile device and its design supports wireless communication and different types of mobile applications. It operates on smart phones, Tablets and Digital mobile devices. 
Ex: Google Android, Apple iOS, Blackberry, Symbian.

27. There are 3 Fundamental Gates in Boolean Algebra. They are AND, OR and NOT.

**The Truth Table of AND Gate**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**The Truth Table of OR Gate**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**The Truth Table of NOT Gate**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

28. **Thunder Bird**

Thunderbird is an E-mail Software which gives the user access to email such as Exchange, Gmail, Hotmail, etc in Ubuntu OS.

**Firefox**

Firefox is a web browser Software which gives the user access to browsing through internet in Ubuntu OS.

29. The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. The switch statement replaces multiple if-else sequence.

**Prepared By : T. Thirumalai M.Sc(CS)., B.Ed.,**
The syntax of the switch statement is:
switch(expression)
{
  case constant 1:
    statement(s);
    break;
  case constant 2:
    statement(s);
    break;
  ....
  default:
    statement(s);
}

30. Array of Strings:
An array of strings is a two-dimensional character array. The size of the first index (rows) denotes the number of strings and the size of the second index (columns) denotes the maximum length of each string. Usually, array of strings are declared in such a way to accommodate the null character at the end of each string. For example, the 2-D array has the declaration:
char Name[6][10];

31. | Constructor          | Destructor                      |
    |----------------------|---------------------------------|
    | The name of the constructor must be same as that of the class. | The destructor has the same name as that of the class prefixed by the tilde character ‘~’. |
    | A constructor can have parameter list. | The destructor cannot have arguments. |
    | The constructor function can be overloaded. | Destructors cannot be overloaded i.e., there can be only one destructor in a class. |

32. Rules for Function Overloading:
1. The overloaded function must differ in the number of its arguments or data types.
2. The return type of overloaded functions are not considered for overloading same data type.
3. The default arguments of overloaded functions are not considered as part of the parameter list in function overloading.

33. When a derived class member function has the same name as that of its base class member function, the derived class member function shadows/hides the base class’s inherited function. This situation is called function overriding and this can be resolved by giving the base class name followed by :: and the member function name.
### 5-Marks

<table>
<thead>
<tr>
<th>SN</th>
<th>Generation</th>
<th>Period</th>
<th>Main Component Used</th>
<th>Merits/Demerits</th>
</tr>
</thead>
</table>
| 1  | First Generation | 1942-1955    | Vacuum tubes        | - Big in size  
- Consumed more power  
- Malfunction due to overheat  
- Machine Language was used |
| 2  | Second Generation| 1955-1964    | Transistors         | - Smaller compared to First Generation  
- Generated Less Heat  
- Consumed less power compared to first generation  
- Punched cards were used  
- First operating system was developed - Batch Processing and Multiprogramming Operating System  
- Machine language as well as Assembly language was used |
| 3  | Third Generation | 1964-1975    | Integrated Circuits (IC) | - Computers were smaller, faster and more reliable  
- Consumed less power  
- High Level Languages were used |
| 4  | Fourth Generation| 1975-1980    | Microprocessor Very Large Scale Integrated Circuits (VLSI) | - Smaller and Faster  
- Microcomputer series such as IBM and APPLE were developed  
- Portable Computers were introduced. |

First Generation Computers - ENIAC, EDVAC, UNIVAC 1  
ENIAC weighed about 27 tons, size 8 feet x 100 feet x 3 feet and consumed around 150 watts of power.

Second Generation Computers - IBM 1401, IBM 1620, UNIVAC 1108

Third Generation Computers - IBM 360 series, Honeywell 6000 series

Prepared By: T. Thirumalai M.Sc(CS), B.Ed.
Read Only Memory (ROM)

Read Only Memory refers to special memory in a computer with pre-recorded data at manufacturing time which cannot be modified. The stored programs that start the computer and perform diagnostics are available in ROMs. ROM stores critical programs such as the program that boots the computer. Once the data has been written onto a ROM chip, it cannot be modified or removed and can only be read. ROM retains its contents even when the computer is turned off. So, ROM is called as a non-volatile memory.

Programmable Read Only Memory (PROM)

Programmable read only memory is also a non-volatile memory on which data can be written only once. Once a program has been written onto a PROM, it remains there forever. Unlike the main memory, PROMs retain their contents even when the computer is turned off. The PROM differs from ROM. PROM is manufactured as a blank memory, whereas a ROM is programmed during the manufacturing process itself. PROM programmer or a PROM burner is used to write data to a PROM chip. The process of programming a PROM is called burning the PROM.

Erasable Programmable Read Only Memory (EPROM)

Erasable Programmable Read Only Memory is a special type of memory which serves as a PROM, but the content can be erased using ultraviolet rays. EPROM retains its contents until it is exposed to ultraviolet light. The ultraviolet light...
clears its contents, making it possible to reprogram the memory. An EPROM differs from a PROM, PROM can be written only once and cannot be erased. EPROMs are used widely in personal computers because they enable the manufacturer to change the contents of the PROM to replace with updated versions or erase the contents before the computer is delivered.

**Electrically Erasable Programmable Read Only Memory (EEPROM)**

Electrically Erasable Programmable Read Only Memory is a special type of PROM that can be erased by exposing it to an electrical charge. Like other types of PROM, EEPROM retains its contents even when the power is turned off. Comparing with all other types of ROM, EEPROM is slower in performance.

35.

<table>
<thead>
<tr>
<th>Logical Gates</th>
<th>Symbol</th>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AND</strong></td>
<td>![AND Symbol]</td>
<td><strong>A</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>OR</strong></td>
<td>![OR Symbol]</td>
<td><strong>A</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>NOT</strong></td>
<td>![NOT Symbol]</td>
<td><strong>A</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

(or)

a) $-98$

98 → 8bit Binary Code → 01100010₂

1’s Complement for -98 → 10011101₂ (interchange 0’s as 1 and 1’s as 0)

2’s Complement for -98 → 10011110₂ (by Add 1 to the 1’s Complement value of -98)

b) $-135$

135 → 8bit Binary Code → 10000111₂

1’s Complement for -135 → 01110000₂ (interchange 0’s as 1 and 1’s as 0)

2’s Complement for -135 → 01111001₂ (by Add 1 to the 1’s Complement value of -135)

Prepared By: T. Thirumalai M.Sc(CS), B.Ed,
Entry Control Loop
A statement or set of statements forms a body of the loop that are executed repetitively. In an entry-controlled loop, first the test-expression is evaluated and if it is nonzero, the body of the loop is executed otherwise the loop is terminated.

There are 2 types Entry control loop. They are While loop and For loop.

Prepared By: T. Thirumalai M.Sc(CS),B.Ed.
**While Loop**

A while loop is a control flow statement that allows the loop statements to be executed as long as the condition is true. The while loop is an entry-controlled loop because the test expression is evaluated before the entering into a loop.

The while loop syntax is:
```
while ( test expression )
{
    Body of the loop;
}
Statement-x;
```

The control flow and flow chart of the while loop is shown below.

In while loop, the test expression is evaluated and if the test expression result is true, then the body of the loop is executed and again the control is transferred to the while loop. When the test expression result is false the control is transferred to statement-x.

```
#include <iostream>
using namespace std;
int main ()
{
    int i=1,sum=0;
    while(i<=10)
    {
        sum=sum+i;
        i++;
    }
    cout<<"The sum of 1 to 10 is "<<sum;
    return 0;
}
```

**Output**
The sum of 1 to 10 is 55
C++ program to sum numbers from 1 to 10 using while loop
In the above program, the integer variable i is initialized to 1 and the variable sum to 0. The while loop checks the condition, i < 10, if the condition is true, the value of i, which is added to sum and i is incremented by 1. Again, the condition i < 10 is checked. Since 2 < 10, 2 is added to the earlier value of sum. This continues until i becomes 11. At this point in time, 11 < 10 evaluates to false and the while loop terminates. After the loop termination, the value of sum is displayed.

37. Call by value Method
This method copies the value of an actual parameter into the formal parameter of the function. In this case, changes made to formal parameter within the function will have no effect on the actual parameter.
#include<iostream>
using namespace std;
void display(int x)
{
    int a=x*x;
    cout<<"\n\nThe Value inside display function (a * a):"<<a;
}
int main()
{
    int a;
    cout<<"\nExample : Function call by value:"
    cout<<"\nEnter the Value for A :";
    cin>>a;
    display(a);
    cout<<"\nThe Value inside main function "<<a;
    return 0;
}
Output :
Example : Function call by value
Enter the Value for A : 5
The Value inside display function (a * a) : 25
The Value inside main function 5
(or)

Advantages of OOP

Re-usability:
“Write once and use it multiple times” you can achieve this by using class.

Redundancy:
Inheritance is the good feature for data redundancy. If you need a same functionality in multiple class you can write a common class for the same functionality and inherit that class to sub class.

Easy Maintenance:
It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.

Prepared By : T.Thirumalai M.Sc(CS),B.Ed.,
**Security:**
Using data hiding and abstraction only necessary data will be provided thus maintains the security of data.

38. **Types of Inheritance**
There are different types of inheritance viz., Single Inheritance, Multiple inheritance, Multilevel inheritance, hybrid inheritance and hierarchical inheritance.

1. **Single Inheritance**
   When a derived class inherits only from one base class, it is known as single inheritance

2. **Multiple Inheritance**
   When a derived class inherits from multiple base classes it is known as multiple inheritance

3. **Hierarchical inheritance**
   When more than one derived classes are created from a single base class, it is known as Hierarchical inheritance.

4. **Multilevel Inheritance**
   The transitive nature of inheritance is itself reflected by this form of inheritance. When a class is derived from a class which is a derived class – then it is referred to as multilevel inheritance.

5. **Hybrid inheritance**
   When there is a combination of more than one type of inheritance, it is known as hybrid inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical, Multilevel and Multiple inheritance.

The following diagram represents the different types of inheritance

![Diagram of different types of inheritance](https://www.trbtnpsc.com/2018/01/11th-revision-test-question-papers-key.html)
The program has error so we can’t get the output.
The Error is
[Error] expected unqualified-id before 'or' token

After changing the Program lines

```cpp
#include<iostream>
#include<stdio.h>
using namespace std;

class P
{
    public:
    P()
    {
        cout<< "\nConstructor of class P ";
    }
    ~P()
    {
        cout<< "\nDestructor of class P ";
    }
};
class Q
{
    public:
    Q()
    {
        cout<<"\nConstructor of class Q ";
    }
    ~Q()
    {
        cout<< "\nDestructor of class Q ";
    }
};
class R
{
    //P obj1,obj2;
    Q obj3;
    public:
    R()
    {
        cout<< "\nConstructor of class R ";
    }
    ~R()
    {
        cout<< "\nDestructor of class R ";
    }
};

int main()
{
    P op;
    //Q oq;
    R oq;
    return 0;
}

**********

Prepared By: T.Thirumalai, M.Sc(CS).,B.Ed.,
Cell: 9750827717

Prepared By : T.Thirumalai M.Sc(CS).,B.Ed.,

https://www.trbtpnc.com/2018/01/11th-revision-test-question-papers-key.html