Programming in C++: Assignment Week 1

Total Marks : 20

August 2, 2017

Question 1

Which special symbol allowed in a variable name? Mark 1

a) !
b) |
c) *
d) _

Answer: d)
Explanation: As per the Syntax of the variable declaration, underscore is the only special symbol allowed in the variable. Refer Slide.

Question 2

Which is the only ternary operator in C? Marks 1

a) ?:
b) &&
c) *=
d) <<

Answer: a) ?:
Explanation: ?: is the only ternary operator in C since it takes 3 operand. Refer Slides

Question 3

Which of the following declarations are correct? Mark 1

a) struct {int a;}
b) struct mystruct {int a;};
c) struct mystruct {int a;}
d) struct mystruct: int a;

Answer: b)
Explanation: As per the Syntax of the language. Refer Slides.
Question 4

Which of the following statement is true about the function `func`? Mark 1

```c
void func(int x, int y) {
    x--; y--;
    return (x+y);
}
```

a) The sum of x and y

b) The sum of the decremented value of x and y

c) returns a pointer to the sum of the decremented value of x and y

d) Compilation Error: return value type does not match the function type

**Answer:** d

**Explanation:** The return type of the function is void, hence an integer value cannot be returned.

Question 5

What value will be printed for `data.c`? Marks 2

```c
#include<stdio.h>
#include <string.h>

int main() {
    union Data {
        int i;
        unsigned char c;
    } data;

    data.i = 89;
    data.c = 'A';
    printf( "%d\n", data.i);
    return 0;
}
```

a) 65

b) 89

c) 0

d) garbage

**Answer:** a

**Explanation:** In union the last assigned value of the variable overwrites the rest of the values depending upon the amount of memory it is allocated. So 1st byte of `data.i` and `data.c` will have same value now. Again When `%d` is used for printing an character value, ASCII code gets converted to integer.
Question 6
What will be the output of the following program? Marks 2

#include <stdio.h>
int main() {
    int i_ = 2, *j_, k_;
    j_ = &i_;
    printf("%d\n", i_**j_*i_+*j_);
    return 0;
}

a) Compilation Error: Erroneous syntax
b) 16
c) 10
d) 8

Answer: c) 10
Explanation: Here Dereference operator (*) has higher priority than multiplication operator (*). So first *j is evaluated and their values are used for multiplication and later for addition: The expression evaluates as: (2 * 2 * 2) + 2

Question 7
What is the output of the following program? Marks 2

#include <stdio.h>
#define func(x, y) x + y/x
int main() {
    int i = -1, j = 2;
    printf("%d\n",func(i + j, 3));
    return 0;
}

a) divide by zero error
b) 0
c) 4
d) -4

Answer: b)
Explanation: x + y/x replaced by i + j + 3/i + j

Question 8
What will be the output of the following program? Marks 2
#include <stdio.h>
int sum(int a, int b, int c) {
    return a*b*c;
}
int main() {
    int (*function_pointer)(int, int, int);
    function_pointer = sum;
    printf("%d", function_pointer(1, 4.5, 5));
    return 0;
}

a) 22.5
b) Compilation Error: Error in function arguments
c) 20
d) Compilation Error: Invalid assignment of sum

Answer: c)
Explanation: function_pointer is a pointer defined for any function with 3 integer parameters and integer return type. The float parameter is implicitly converted to int

Question 9
Fill the blank by Choosing the correct option(s)to concatenate strings str1 and str2 to form str3? Marks 2
#include <iostream>
#include <string>
using namespace std;

int main(void) {
    string str1 = "I Love to ";
    string str2 = "Cycle";

    string str3 = _______________________
    cout << str3;
    return 0;
}
Output: I Love to Cycle

a) str1+str2
b) strcat(str1,str2)
c) str1.append(str2)
d) strcat(strcpy(str3,str1),str2)

Answer: a) c)
Explanation: str1 and str 2 are two string type variables, operations possible for concatenation are str1+str2 (String is a stl, hence has + operator overloaded) and str1.append(str2) to append strings.
Question 10

What will be the output of the following program? *Marks 2*

```cpp
#include <iostream>
#include <algorithm>
using namespace std;
bool srt (int i, int j) {
    return (i < j);
}
int main() {
    int data[] = {52, 76, 19, 5, 10, 100, 56, 98, 17};
    sort (data + 1, data + 5, srt);
    for (int i = 0; i < 7; i++)
        cout << data[i] << " ";
    return 0;
}
```

a) 5 10 19 52 56 76 98 100 17  
b) 5 10 19 52 76 100 56 98 17  
c) 52 5 10 19 76 100 56 98 17  
d) 52 5 10 19 76 100 56  

**Answer:** d)  
**Explanation:** The whole array is not passed for sorting, only from index 1 (data + 1, i.e 0 + 1) to index 5 (data + 5, i.e 0 + 5), i.e 3 elements, 76, 19, 5,10

Question 11

What will be the output of the following program? *Marks 2*

```cpp
#include<iostream>  
#include<string.h>  
#include<stack>  
using namespace std;
int main() {
    char str[19]= "Accessing";
    stack<char> s;
    for(int i = 0; i < strlen(str); i++)
        s.push(str[i]);
    for(int i = 0; i < strlen(str) - 1; i++) {
        s.top(); s.pop();
        cout << s.top();
    }
    return 0;
}
```

a) gnisseccA  
b) nisseccA  
c) gnissecc
d) nisseccAnisseccA  

**Answer:** b)  

**Explanation:** When 'Accessing' is pushed to stack, the element on the top is g (gnisseccA), which is popped and then the next element is displayed till str - 1

**Question 12**

Fill up the blanks for A# and B# below: *Marks 2*

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    cout << "Enter the no. of elements: ";
    int count, j, sub=0;
    cin >> count;
    _______________ A# // Declare with Default size
    _______________ B# // Change the size to the required amount
    for(int i = 0; i < v.size(); i++) {
        v[i] = i;
        sub -= v[i];
    }
    cout << "Array Sum: " << sub<< endl;
    return 0;
}
```

a) A#: vector <int> v;  
   B#: v.resize(count);

b) A#: vector <int> v(count);  
   B#: v.resize(count);

c) A#: vector <int> v(count);  
   B#: v.size(count);

d) A#: vector <int> v;  
   B#: v.size(count);

**Answer:** a)  

**Explanation:** As per syntax, using resize operator