1.1 45
1.2 Twice
1.3 0
1.4 6
1.5 7
1.6 Name.size()
1.7 0
1.8 Classic basic primitive types may include:

Character (character, char);

Integer (integer, int, short, long, byte) with a variety of precisions;

Floating-point number (float, double, real, double precision);

Boolean, logical values true and false

1.9 SAfrica

1.10 #include <iostream>

Question 2)

2.1) To add all the even numbers in the array to sum variable, and display sum

2.2) To accept values to variable a and b, swap the values in the variables and display the variables.

Question 3)

3.1)

```
if(level == 1 || level == 2)
      cout<<"Basic";
   else if(level == 3)
      cout<<"Intermediate";
   else if(level == 4)
      cout<<"Advance";
   else
      cout<<"Not a valid level";
```

3.2)
A **prime number** is a whole **number** greater than 1, whose only two whole-**number** factors are 1 and itself. The first few **prime numbers** are 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29.

```
int i=2, j;
```

```
bool prime = true;
while(i < 10)
{
   prime = true;
   j = 2;
   while(j<i)
   {
      if((i%j) == 0)
         prime = false;
      j++;
   }
   if(prime)
      cout<<i<<endl;
   i++;
}
```

4.1)
A named memory location used to store data of a specific datatype

4.2)
Global variable are variables which are declared outside all function and can be accessed by all the functions within that program.

Local variable are variable which are declared within a function or block of code(e.g for_loop) and are visible within that function or block.

4.3)
float f1;

4.4)
A function is a group of statements that together perform a task. A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.

4.5)Refer to Appendix D of the leaner guide, page 376
getline(istream, string &, char)
srand(int)
int rand( )
char tolower(char)

4.6)
Acutal parameters are values in a function calling statement that are passed to the function being called.

Formal are parameter are variables in the function definition that are used to receive values for that function.

4.7)
A value parameter is a function variable that receives a copy of the passed value and store it in its own memory location.

A reference parameter is a function variable that receives a reference (memory address) of the value been passed to it, it use the same memory location with the argument in the calling statement.

4.8)
float calcInterest(float principalP, float rateP, int yearsP)

4.9.1)
result, num3, num4
4.9.2)
product, num1, num2
4.9.3)
result
4.9.4)
Num3, num4

Question 5)

5.1)
```
struct Employee{

   int staffNo;
   string DeptID;
   float AnnualSalary;
};
```

5.2)

```
Employee ABC[50];
```

5.3)

```
for(int i=0; i<5; i++)
   {
      cout<<"Enter staff no. of the employee"<<endl;
      cin>>ABC[i].staffNo;

      cout<<"Enter department ID of the employee"<<endl;
      cin>>ABC[i].DeptID;

      cout<<"Enter annual salary of the employee"<<endl;
      cin>>ABC[i].AnnualSalary;
```

```
    }
```

Question 6)

6.1)

```
int intArray[10];
   int total=0;
   float avg;
   for(int i=0; i<10; i++)
   {
      cout<<"Enter integer number"<<endl;
      cin>>intArray[i];
      total += intArray[i];
   }
   avg = float(total / 10);

   cout<<avg;
```

6.2)

```
#include <iostream>
#include<string>
using namespace std;

const int size_of_array = 10;

bool checkNumber(const int arrP[], int & posP, int numP)
{
   for(int i = 0; i < size_of_array; i++){
      if(arrP[i]==num)
      {
         posP = i;
         return true;
      }
   }
   return false;
}
int main()
{
   int arr[size_of_array], pos=-1, num;
   bool yesNo;

   cout << "Enter element of the array" << endl;

   for(int i = 0; i < size_of_array; i++){
      cin >> arr[i];
```

```
    }
     cout << "Enter an integer number" << endl;
     cin>>num;

     yesNo = checkNumber(arr, pos, num);

     if(yesNo)
       cout<<"The integer appears at position "<<pos<<" of the array";
     else
       cout<<"The integer is not found in the array";

   return 0;
}
```

6.2.1)

```
bool checkNumber(const int arrP[], int & posP, int num)
{
   for(int i = 0; i < size_of_array; i++){
      if(arrP[i]==num)
      {
         posP = i;
         return true;
      }
   }
   return false;
}
```

6.2.2)

```
    yesNo = checkNumber(arr, pos, num);
```

6.2.3)

```
cout<<"The integer appears at position "<<pos<<" of the array";
```

6.3)

```
#include <iostream>
#include<string>
using namespace std;

int main()
{
   string fullname, fname, lname;
   int spacePos;

   cout << "Enter full name" << endl;
```

```
    getline(cin, fullname, '\n');

    spacePos = fullname.find(" ");

    fname = fullname.substr(0, spacePos);
    lname = fullname.substr(spacePos + 1);

    cout<<lname<<", "<< fname<<endl;


    return 0;
}
```

6.4)

```
#include <iostream>
#include<string>
using namespace std;

int main()
{
    const int row_size = 4;
    const int col_size = 3;
    int arr[row_size][col_size] = {7,17,8,52,4,15,25,6,14,30,2,10};
    int val5;
    for (int i = 0; i < row_size; i++)
    {
        for (int j = 0; j < col_size; j++)
        {
            if((arr[i][j]%5) == 0)
                arr[i][j] += 5;
            cout<<arr[i][j]<< ", ";
        }

    }


    return 0;
}
```

Answer:

```
    for (int i = 0; i < row_size; i++)
    {
        for (int j = 0; j < col_size; j++)
        {
            if((arr[i][j]%5) == 0)
                arr[i][j] += 5;
        }
```

```
   }
```

Question 7)

7.1)

```
bool p = true, q = false;
   int x = 5;
   if(!(p && !(p || q)))
      ++x;
   x++;

   cout<<x<<endl;
```

Answer is: 7

7.2)

```
#include <iostream>
#include<string>
using namespace std;

void multiply1(int & p, int & q)
{
   int r = p * q++;
}
void multiply2(int & p, int & q)
{
   int r = ++p * q;
}
int main()
{
   int p = 2, q = 3;
   multiply1(p,q);
   multiply2(p,q);
   cout<<p * q<<endl;
   return 0;
}
```

Answer is: 12

7.3)

```
#include <iostream>
#include<string>
using namespace std;

void funcl(int & x, int & y, int z)
{
```

```
    z += (++x) + (y++);
}
void func2(int x, int y, int &sum)
{
    sum = x + y;
}
int main()
{
  int a = 4; b = 5, c = 0;
  funcl(a,b,c);
  cout<<a<<b<<c<<endl;
  func2(a,b,c);
  cout<<a<<b<<c<<endl;
}
```

Question 8)

```
#include <iostream>
#include<string>
using namespace std;

struct Sales
{
    int salesID;
    float salesAmount[4];
};
void calcSales(Sales & salesP)
{
    float totalSales=0;
    for(int i =0; i<4; i++)
    {
        totalSales+=salesP.salesAmount[i];
    }
    cout<<totalSales<<endl;
}
int main()
{
    Sales sale1;

    cout << "Enter sales person ID" << endl;
    cin >> sale1.salesID;


        cout << "Enter sales person's 4 quarterly amounts " << endl;
        for(int i = 0; i < 4; i++){

        cin >> sale1.salesAmount[i];
    }
```

```
   calcSales(sale1);

   return 0;
}
```

8.1)

```
struct Sales
{
   string salesID;
   float salesAmount[4];
};
```

8.2)

```
Sales sale1;
```

8.3)

```
void calcSales(Sales & salesP)
{
   float totalSales=0;
   for(int i =0; i<4; i++)
   {
      totalSales+=salesP.salesAmount[i];
   }
   cout<<totalSales<<endl;
}
```