

Tutorial 33 - Dynamic Initialization of Objects Using Constructors

Key Concepts

1. **Dynamic Initialization:**

- Initialization of objects during runtime.
- Useful when data comes in different formats or is provided at runtime.

2. **Constructor Overloading:**

- Constructors with different parameter types or counts handle initialization based on provided arguments.

Code Example and Explanation

Code Snippet 1: Class Declaration

```
1 #include<iostream>
2 using namespace std;
3 class BankDeposit {
4     int principal, years;
5     float interestRate, returnValue;
6 public:
7     BankDeposit() {} // Default Constructor
8     BankDeposit(int p, int y, float r); // Float interest rate
9     BankDeposit(int p, int y, int r);   // Integer interest rate
10    void show();
11 };
12
```

Key Points:

1. **Data Members:**

- `principal`, `years`: Loan details.
- `interestRate`, `returnValue`: Interest rate and total return.

2. **Constructors:**

- **Default Constructor:** No initialization.
- **Overloaded Constructors:**
 - One takes `float` interest rate (`r`).
 - Another takes `int` interest rate (`r`).

3. **Function** `show`: Prints object details.

Code Snippet 2: Constructor Definitions

```
1 BankDeposit::BankDeposit(int p, int y, float r) {
2     principal = p;
3     years = y;
4     interestRate = r;
5     returnValue = principal;
6     for (int i = 0; i < y; i++) {
7         returnValue *= (1 + interestRate);
8     }
9 }
10 BankDeposit::BankDeposit(int p, int y, int r) {
```

```

11     principal = p;
12     years = y;
13     interestRate = float(r) / 100;
14     returnValue = principal;
15     for (int i = 0; i < y; i++) {
16         returnValue *= (1 + interestRate);
17     }
18 }
19 void BankDeposit::show() {
20     cout << "\nPrincipal amount was " << principal
21         << ". Return value after " << years
22         << " years is " << returnValue << endl;
23 }
24

```

Explanation:

1. Constructor with Float `r`:

- Calculates compound interest using `r` as a percentage (e.g., `0.05`).

2. Constructor with Integer `r`:

- Converts `r` to a float percentage (e.g., `5` → `0.05`).

3. Compound Interest:

- Iterates for `years`, applying interest to `returnValue`.

Code Snippet 3: Main Program

```

1  int main() {
2      BankDeposit bd1, bd2;
3      int p, y, R;
4      float r;
5      cout << "Enter the value of p, y, and r: ";
6      cin >> p >> y >> r;
7      bd1 = BankDeposit(p, y, r); // Calls float constructor
8      bd1.show();
9      cout << "Enter the value of p, y, and R: ";
10     cin >> p >> y >> R;
11     bd2 = BankDeposit(p, y, R); // Calls int constructor
12     bd2.show();
13     return 0;
14 }
15

```

Execution:

1. Dynamic Input:

- `p`, `y`, and `r` or `R` are entered at runtime.

2. Constructor Calls:

- Float `r`: Calls constructor with `float` parameter.
- Integer `R`: Calls constructor with `int` parameter.

Short Notes for Notebook

Dynamic Initialization of Objects

1. Definition:

- Initializing objects at runtime using constructors.
- Handles data in different formats (e.g., float, int).

2. Class Example:

```
1 class BankDeposit {
2     int principal, years;
3     float interestRate, returnValue;
4 public:
5     BankDeposit(); // Default Constructor
6     BankDeposit(int p, int y, float r); // Float rate
7     BankDeposit(int p, int y, int r);   // Integer rate
8     void show();
9 };
10
```

3. Key Points:

- Overloaded constructors calculate compound interest.
- Float **r**: Rate as a decimal (e.g., 0.05).
- Integer **r**: Rate as a whole number (e.g., 5 → 0.05).

Main Program Example

```
1 int main() {
2     BankDeposit bd1, bd2;
3     int p, y, R;
4     float r;
5     cout << "Enter the value of p, y, and r: ";
6     cin >> p >> y >> r;
7     bd1 = BankDeposit(p, y, r); // Float constructor
8     bd1.show();
9     cout << "Enter the value of p, y, and R: ";
10    cin >> p >> y >> R;
11    bd2 = BankDeposit(p, y, R); // Int constructor
12    bd2.show();
13    return 0;
14 }
15
```

Output Example

```
1 Enter the value of p, y, and r:
2 100 1 0.05
3 Principal amount was 100. Return value after 1 years is 105.
4 Enter the value of p, y, and R:
5 100 1 5
6 Principal amount was 100. Return value after 1 years is 105.
```